# Java Programming

**Question 01:** Which company developed Java?

a) Microsoft
b) Oracle
c) Apple
d) Sun Microsystem

**Answer : D**

Explanation: None.

**Question 02:** Which was the original name of Java at the time of development?

a) CupofCoffee
b) Oak
c) Sun++
d) Gava

**Answer : B**

Explanation: None.

**Question 03:** Java is ?

a) Method Oriented Programming
b) Object Oriented Programming
c) Pure Object Oriented Programming
d) Structural Oriented Programming

**Answer : C**

Explanation: None.



**Question 04:** Which company owns Java now?

a) Sun MicroSystem
b) IBM
c) Apple
d) Oracle

**Answer : D**

Explanation: None.

**Question 05:** After compilation of a Java program which code generated?

a) Executable code (.exe)
b) Assembly code (.asm)
c) Object Code (.obj)
d) Byte Code (.class)

**Answer : D**

Explanation: After compilation class file (Byte code) generated.

**Question 06:** Which declarations are required in a Java program?

a) There should be a main function
b) There should be a class
c) There should be a class and a main function
d) None of these

**Answer : C**

Explanation: There should be a class and a main function

**Question 07:** Which is not a valid type of variable?

a) Global variable
b) Local variable
c) Class variable
d) Instance variable

**Answer : A**

Explanation: None.



**Question 08:** Which is a primitive data type?

a) String
b) Character
c) Float
d) byte

**Answer : D**

Explanation: String, Character and Float are classes but byte is primitive data type.

**Question 09:** What will be the output of following program?

```
public class Prg
{
    public static void main(String args[])
    {
        System.out.print(20+ 1.34f + "A" + "B");
    }
}
```

a) 201.34AB
b) 201.34fAB
c) 21.34AB
d) Error

**Answer : C**

Explanation: None.

**Question 10:** What will be the output of following program?

```
public class Prg
{
    public static void main(String[] args)
    {
            char [] str={'i','n','c','l','u','d','e','h','e','l','p'};
            System.out.println(str.toString());
    }
}
```

a) includehelp
b) Error
c) [C@19e0bfd (Memory Address)
d) NULL

**Answer : C**

Explanation: str is a character array, if you try to print str.toString() it will not converted to string because str is an object of character array that will print an address in string format

**Question 11:** What should be the name of java program file containing this program?

```
public class MyPrg
{
    public static void main(String args[])
    {
        System.out.print("IncludeHelp");
    }
}
```

a) MyPrg.class
b) MyPrg.java
c) MyPrg
d) Any file name with java extension

**Answer : B**

Explanation: In this program class MyPrg is public, so we can not take any file name, we must save this program by MyPrg.java file name otherwise Compilation error is occurred.

**Question 12:** What is byte code in Java?

a) It is another name for java source file that contain the information about the hardware.
b) It is a binary code generated by the Java Virtual Machine for operating system.
c) It is an intermediate code generated by the java compiler for Java Virtual Machine.
d) None of these

**Answer : C**

Explanation: It is an intermediate code generated by the java compiler for Java Virtual Machine.

**Question 13:** What is Garbage Collection (GC) in the Java?

a) It is a thread to free the memory of de-referenced objects automatically.
b) It is a method to collect the garbage values of variables.
c) It is a package included in program to free the memory occupied by objects.
d) None of these

**Answer : A**

Explanation: None.

**Question 14:** What will be the output of following program?

```
class Prg
{
    public static void main(String args[])
    {
        const int a=10;
        System.out.println(a);
    }
}
```

a) 10
b) a
c) Unprintable Character
d) CompileTime Error

**Answer : D**

Explanation: JAVA does not support the const keyword, instead of const, final keyword is used.

# Data Types

**Question 15:** What is the range of short data type in Java?

a) -128 to 127
b) -32768 to 32767
c) -2147483648 to 2147483647
d) None of the mentioned

**Answer : B**

Explanation: Short occupies 16 bits in memory. Its range is from -32768 to 32767.

**Question 16:** What is the range of byte data type in Java?

a) -128 to 127
b) -32768 to 32767
c) -2147483648 to 2147483647
d) None of the mentioned

**Answer : A**

Explanation: Byte occupies 8 bits in memory. Its range is from -128 to 127.

**Question 17:** In Java, Numbers are considered as ?

a) Integer
b) int
c) long
d) Float

**Answer : A**

Explanation: None.

**Question 18:** An expression involving byte, int, and literal numbers is promoted to which of these?

a) int
b) long
c) byte
d) float

**Answer : A**

Explanation: An expression involving bytes, ints, shorts, literal numbers, the entire expression is promoted to int before any calculation is done.

**Question 19:** Which of these literals can be contained in float data type variable?

a) -1.7e+308
b) -3.4e+038
c) +1.7e+308
d) -3.4e+050

**Answer : B**

Explanation: Range of float data type is -(3.4e38) To +(3.4e38)

**Question 20:** Which data type value is returned by all transcendental math functions?

a) int
b) float
c) double
d) long

**Answer : C**

Explanation: None.

**Question 21:** What is the output of this program?

```
class average
{
    public static void main(String args[])
    {
        double num[] = {5.5, 10.1, 11, 12.8, 56.9, 2.5};
        double result;
        result = 0;
        for (int i = 0; i < 6; ++i)
            result = result + num[i];
        System.out.print(result/6);
    }
}
```

a) 16.34
b) 16.566666644
c) 16.46666666666667
d) 16.466666666666667

**Answer : C**

Explanation: None.

**Question 22:** What will be the output of these statement?

```
class output
{
    public static void main(String args[])
    {
        double a, b,c;
        a = 3.0/0;
        b = 0/4.0;
        c=0/0.0;
        System.out.println(a);
        System.out.println(b);
        System.out.println(c);
    }
}
```

a) Infinity
b) 0.0
c) NaN
d) all of the mentioned

**Answer : D**

Explanation: For floating point literals, we have constant value to represent (10/0.0) infinity either positive or negative and also have NaN (not a number for undefined like 0/0.0), but for the integral type, we don't have any constant that's why we get an arithmetic exception.

**Question 23:** What is the output of this program?

```
class increment
{
    public static void main(String args[])
    {
        int g = 3;
        System.out.print(++g * 8);
    }
}
```

a) 25
b) 24
c) 32
d) 33

**Answer : C**

Explanation: Operator ++ has more preference than *, thus g becomes 4 and when multiplied by 8 gives 32.

**Question 24:** What is the output of this program?

```
class area
{
    public static void main(String args[])
    {
        double r, pi, a;
        r = 9.8;      pi = 3.14;
        a = pi * r * r;
        System.out.println(a);
    }
}
```

a) 301.5656
b) 301
c) 301.56
d) 301.56560000

**Answer : A**

Explanation: None

**Question 25:** What is the numerical range of a char data type in Java?

a) -128 to 127
b) 0 to 256
c) 0 to 32767
d) 0 to 65535

**Answer : D**

Explanation: Char occupies 16-bit in memory, so it supports 2^16 i:e from 0 to 65535.

**Question 26:** Which of these coding types is used for data type characters in Java?

a) ASCII
b) ISO-LATIN-1
c) UNICODE
d) None of the mentioned

**Answer : C**

Explanation: Unicode defines fully international character set that can represent all the characters found in all human languages. Its range is from 0 to 65536.

**Question 27:** Which of these values can a boolean variable contain?

a) True & False
b) 0 & 1
c) Any integer value
d) true

**Answer : A**

Explanation: None.

**Question 28:** Which of these occupy first 0 to 127 in Unicode character set used for characters in Java?

a) ASCII
b) ISO-LATIN-1
c) None of the mentioned
d) ASCII and ISO-LATIN1

**Answer : D**

Explanation: First 0 to 127 character set in Unicode are same as those of ISO-LATIN-1 and ASCII.

**Question 29:** Which one is a valid declaration of a boolean?

a) boolean b1 = 1;
b) boolean b2 = 'false';
c) boolean b3 = false;
d) boolean b4 = 'true'

**Answer : C**

Explanation: Boolean can only be assigned true or false literals

**Question 30:** What is the output of this program?

```
class array_output
{
   public static void main(String args[])
   {
        char array_variable [] = new char[10];
       for (int i = 0; i < 10; ++i)
        {
            array_variable[i] = 'i';
            System.out.print(array_variable[i] + "" );
            i++;
        }
   }
}
```

a) i i i i i
b) 0 1 2 3 4
c) i j k l m
d) None of the mentioned

**Answer : A**

Explanation: None.

**Question 31:** What is the output of this program?

```
class mainclass
{
   public static void main(String args[])
   {
      char a = 'A';
      a++;
      System.out.print((int)a);
   }
}
```

a) 66
b) 67
c) 65
d) 64

**Answer : A**

Explanation: ASCII value of 'A' is 65, on using ++ operator character value increments by one.

**Question 32:** What is the output of this program?

```
class mainclass
{
   public static void main(String args[])
   {
      boolean var1 = true;
      boolean var2 = false;
       if (var1)
          System.out.println(var1);
       else
          System.out.println(var2);
   }
}
```

a) 0
b) 1
c) true
d) false

**Answer : C**

Explanation: None.

**Question 33:** What is the output of this program?

```
class booloperators
{
   public static void main(String args[])
   {
      boolean var1 = true;
      boolean var2 = false;
      System.out.println((var1 & var2));
   }
}
```

a) 0
b) 1
c) true
d) false

**Answer : D**

Explanation: boolean '&' operator always returns true or false. var1 is defined true and var2 is defined false hence their '&' operator result is false.

**Question 34:** What is the output of this program?

```
class asciicodes
{
    public static void main(String args[])
    {
      char var1 = 'A';
      char var2 = 'a';
      System.out.println((int)var1 + " " + (int)var2);
    }
}
```

a) 162
b) 65 97
c) 67 95
d) 66 98

**Answer : B**

Explanation: ASCII code for 'A' is 65 and for 'a' is 97.

**Question 35:** What is the order of variables in Enum?

a) Ascending order
b) Descending order
c) Random order
d) depends on the order() method

**Answer : A**

Explanation: The compareTo() method is implemented to order the variable in ascending order.

**Question 36:** Can we create an instance of Enum outside of Enum itself?

a) True
b) False

**Answer : B**

Explanation: Enum does not have a public constructor.

**Question 37:** What is the output of this program?

```
enum Season {
                WINTER, SPRING, SUMMER, FALL
              };
System.out.println(Season.WINTER.ordinal());
```

a) 0
b) 1
c) 2
d) 3

**Answer : A**

Explanation: ordinal() method provides number to the variables defined in Enum.

**Question 38:** If we try to add Enum constants to a TreeSet, what sorting order will it use?

a) Sorted in the order of declaration of Enums
b) Sorted in alphabetical order of Enums
c) Sorted based on order() method
d) Sorted in descending order of names of Enums

**Answer : A**

Explanation: Tree Set will sort the values in the order in which Enum constants are declared.

**Question 39:** What is the output of below code snippet?

```
class A
{ }
enum Enums extends A
{
   ABC, BCD, CDE, DEF;
}
```

a) Runtime Error
b) Compilation Error
c) It runs successfully
d) EnumNotDefined Exception

**Answer : B**

Explanation: Enum types cannot extend class.

**Question 40:** What is the output of below code snippet?

```
enum Levels
{
   private TOP,
   public MEDIUM,
   protected BOTTOM;
}
```

a) Runtime Error
b) EnumNotDefined Exception
c) It runs successfully
d) Compilation Error

**Answer : D**

Explanation: Enum cannot have any modifiers. They are public, static and final by default.

**Question 41:** What is the output of below code snippet?

```
enum Enums
{
   A, B, C;
   private Enums()
   {
      System.out.println(10);
   }
}
public class MainClass
{
   public static void main(String[] args)
   {
      Enum en = Enums.B;
   }
}
```

a) 10
10
10
b) Compilation Error
c) 10
10
d) Runtime Exception

**Answer : A**

Explanation: The constructor of Enum is called which prints 10.

**Question 42:** Which method returns the elements of Enum class?

a) getEnums()
b) getEnumConstants()
c) getEnumList()
d) getEnum()

**Answer : B**

Explanation: getEnumConstants() returns the elements of this enum class or null if this Class object does not represent an enum type.

**Question 43:** Which class does all the Enums extend?

a) Object
b) Enums
c) Enum
d) EnumClass

**Answer : C**

Explanation: All enums implicitly extend java.lang.Enum. Since Java does not support multiple inheritance, an enum cannot extend anything else.

**Question 44:** Are enums are type-safe?

a) True
b) False

**Answer : A**

Explanation: Enums are type-safe as they have own name-space.

**Question 45:** Which of the following is the advantage of BigDecimal over double?

a) Syntax
b) Memory usage
c) Garbage creation
d) Precision

**Answer : D**

Explanation: BigDecimal has unnatural syntax, needs more memory and creates a great amount of garbage. But it has a high precision which is useful for some calculations like money.

**Question 46:** Which of the below data type doesn't support overloaded methods for +,-,* and /?

a) int
b) float
c) double
d) BigDecimal

**Answer : D**

Explanation: int, float, double provide overloaded methods for +,-,* and /. BigDecimal does not provide these overloaded methods.

**Question 47:** What is the output of below code snippet?

```
double a = 0.02;
double b = 0.03;
double c = b - a;
System.out.println(c);
BigDecimal _a = new BigDecimal("0.02");
BigDecimal _b = new BigDecimal("0.03");
BigDecimal _c = b.subtract(_a);
System.out.println(_c);
```

a) 0.009999999999999998
   0.01
b) 0.01
   0.009999999999999998
c) 0.01
   0.01
d) 0.009999999999999998
   0.009999999999999998

**Answer : A**

Explanation: BigDecimal provides more precision as compared to double. Double is faster in terms of performance as compared to BigDecimal.

**Question 48:** What is the base of BigDecimal data type?

a) Base 2
b) Base 8
c) Base 10
d) Base e

**Answer : C**

Explanation: A BigDecimal is n*10^scale where n is an arbitrary large signed integer. Scale can be thought of as the number of digits to move the decimal point to left or right.

**Question 49:** What is the limitation of toString() method of BigDecimal?

a) There is no limitation
b) toString returns null
c) toString returns the number in expanded form
d) toString uses scientific notation

**Answer : D**

Explanation: toString() of BigDecimal uses scientific notation to represent numbers known as canonical representation. We must use toPlainString() to avoid scientific notation.

**Question 50:** Which of the following is not provided by BigDecimal?

a) scale manipulation
b) + operator
c) rounding
d) hashing

**Answer : B**

Explanation: toBigInteger() converts BigDecimal to a BigInteger.toBigIntegerExact() converts this BigDecimal to a BigInteger by checking for lost information.

**Question 51:** BigDecimal is a part of which package?

a) java.lang
b) java.math
c) java.util
d) java.io

**Answer : B**

Explanation: BigDecimal is a part of java.math. This package provides various classes for storing numbers and mathematical operations.

**Question 52:** What is BigDecimal.ONE?

a) wrong statement
b) custom defined statement
c) static variable with value 1 on scale 10
d) static variable with value 1 on scale 0

**Answer : D**

Explanation: BigDecimal.ONE is a static variable of BigDecimal class with value 1 on scale 0.

**Question 53:** Which class is a library of functions to perform arithmetic operations of BigInteger and BigDecimal?

a) MathContext
b) MathLib
c) BigLib
d) BigContext

**Answer : A**

Explanation: MathContext class is a library of functions to perform arithmetic operations of BigInteger and BigDecimal.

**Question 54:** What is the output of below code snippet?

```
public class AddDemo
{
        public static void main(String args[])
        {
                BigDecimal b = new BigDecimal("23.43");
                BigDecimal br = new BigDecimal("24");
                BigDecimal bres = b.add(new BigDecimal("450.23"));
                System.out.println("Add: "+bres);
                MathContext mc= new MathContext(2, RoundingMode.DOWN);
                BigDecimal bdecMath = b.add(new BigDecimal("450.23"), mc);
                System.out.println("Add using MathContext: "+bdecMath);
        }
}
```

a) Compilation failure
b)
Add: 684.66
Add using MathContext: 6.8E+2
c)
Add 6.8E+2
Add using MathContext: 684.66
d) Runtime exception

**Answer : A**

Explanation: add() adds the two numbers, MathContext provides library for carrying out various arithmetic operations.

# Data Type-Date, TimeZone

**Question 55:** How to format date from one form to another?

a) SimpleDateFormat
b) DateFormat
c) SimpleFormat
d) DateConverter

**Answer : A**

Explanation: SimpleDateFormat can be used as Date now = new Date();
SimpleDateFormat sdf = new SimpleDateFormat ("yyyy-mm-dd'T'hh:MM:ss");
String nowStr = sdf.format(now);
System.out.println("Current Date: " + );

**Question 56:** How to convert Date object to String?

a) SimpleDateFormat sdf = new SimpleDateFormat("yyyy-mm-dd");
sdf.parse(new Date());
b) SimpleDateFormat sdf = new SimpleDateFormat("yyyy-mm-dd");
sdf.format(new Date());
c) SimpleDateFormat sdf = new SimpleDateFormat("yyyy-mm-dd");
new Date().parse();
d) SimpleDateFormat sdf = new SimpleDateFormat("yyyy-mm-dd");
new Date().format();

**Answer : B**

Explanation: SimpleDateFormat takes a string containing pattern. sdf.format converts the Date object to String.

**Question 57:** How to convert a String to a Date object?

a) SimpleDateFormat sdf = new SimpleDateFormat("yyyy-mm-dd");
sdf.parse(new Date());
b) SimpleDateFormat sdf = new SimpleDateFormat("yyyy-mm-dd");
sdf.format(new Date());
c) SimpleDateFormat sdf = new SimpleDateFormat("yyyy-mm-dd");
new Date().parse();
d) SimpleDateFormat sdf = new SimpleDateFormat("yyyy-mm-dd");
new Date().format();

**Answer : A**

SimpleDateFormat takes a string containing pattern. sdf.parse converts the String to Date object.

**Question 58:** Is SimpleDateFormat thread safe?

a) True
b) False

**Answer : B**

Explanation: SimpleDateFormat is not thread safe. In the multithreaded environment we need to manage threads explicitly.

**Question 59:** What is the replacement of joda time library in java 8?

a) java.time (JSR-310)
b) java.date (JSR-310)
c) java.joda
d) java.jodaTime

**Answer : A**

Explanation: In java 8,we are asked to migrate to java.time (JSR-310) which is a core part of the JDK which replaces joda library project.

**Question 60:** How is Date stored in database?

a) java.sql.Date
b) java.util.Date
c) java.sql.DateTime
d) java.util.DateTime

**Answer : A**

Explanation: java.sql.Date is the datatype of Date stored in database.

**Question 61:** What does LocalTime represent?

a) Date without time
b) Time without Date
c) Date and Time
d) Date and Time with timezone

**Answer : B**

Explanation: LocalTime of joda library represents time without date.

**Question 62:** How to get difference between two dates?

a) long diffInMilli = java.time.Duration.between(dateTime1, dateTime2).toMillis();
b) long diffInMilli = java.time.difference(dateTime1, dateTime2).toMillis();
c) Date diffInMilli = java.time.Duration.between(dateTime1, dateTime2).toMillis();
d) Time diffInMilli = java.time.Duration.between(dateTime1, dateTime2).toMillis();

**Answer : A**

Explanation: Java 8 provides a method called between which provides Duration between two times.

**Question 63:** How to get UTC time?

a) Time.getUTC();
b) Date.getUTC();
c) Instant.now();
d) TimeZone.getUTC();

**Answer : C**

Explanation: In java 8, Instant.now() provides current time in UTC/GMT.

# Literals & Variables

**Question 64:** Which of these is long data type literal?

a) 0x99fffL
b) ABCDEFG
c) 0x99fffa
d) 99671246

**Answer : A**

Explanation: Data type long literals are appended by an upper or lowercase L. 0x99fffL is hexadecimal long literal.

**Question 65:** Which of these can be returned by the operator &?

a) Integer
b) Boolean
c) Character
d) Integer or Boolean

**Answer : D**

Explanation: We can use binary ampersand operator on integers/chars (and it returns an integer) or on booleans (and it returns a boolean).

**Question 66:** Literals in java must be appended by which of these?

a) L
b) l
c) D
d) L and l

**Answer : D**

Explanation: Data type long literals are appended by an upper or lowercase L.

**Question 67:** Literal can be of which of these data types?

a) integer
b) float
c) boolean
d) all of the mentioned

**Answer : D**

Explanation: None.

**Question 68:** Which of these cannot be used for a variable name in Java?

a) identifier
b) keyword
c) identifier & keyword
d) none of the mentioned

**Answer : B**

Explanation: Keywords are specially reserved words which cannot be used for naming.

**Question 69:** What is the output of this program?

```
class evaluate
{
   public static void main(String args[])
   {
        int a[] = {1,2,3,4,5};
       int d[] = a;
       int sum = 0;
       for (int j = 0; j < 3; ++j)
             sum += (a[j] * d[j + 1]) + (a[j + 1] * d[j]);
       System.out.println(sum);
   }
}
```

a) 38
b) 39
c) 40
d) 41

**Answer : C**

Explanation: None.

**Question 70:** What is the output of this program?

```
class array_output
{
   public static void main(String args[])
   {
        int array_variable[] = new int[10];
        for (int i = 0; i < 10; ++i)
       {
          array_variable[i] = i/2;
          array_variable[i]++;
          System.out.print(array_variable[i] + " ");
          i++;
       }
   }
}
```

a) 0 2 4 6 8
b) 1 2 3 4 5
c) 0 1 2 3 4 5 6 7 8 9
d) 1 2 3 4 5 6 7 8 9 10

**Answer : B**

Explanation: When an array is declared using new operator then all of its elements are initialized to 0 automatically. for loop body is executed 5 times as whenever controls comes in the loop i value is incremented twice, first by i++ in body of loop then by ++i in increment condition of for loop.

**Question 71:** What is the output of this program?

```
class variable_scope
{
   public static void main(String args[])
   {
      int x=5;
      {
            int y = 6;
            System.out.print(x + " " + y);
      }
      System.out.println(x + " " + y);
   }
}
```

a) 5 6 5 6
b) 5 6 5
c) Runtime error
d) Compilation error

**Answer : D**

Explanation: Second print statement doesn't have access to y , scope y was limited to the block defined after initialization of x.

**Question 72:** Which of these is an incorrect string literal?

a) "Hello World"
b) "Hello\nWorld"
c) "\"Hello World\""
d)
"Hello
world"

**Answer : D**

Explanation: all string literals must begin and end in the same line.

**Question 73:** What is the output of this program?

```
class dynamic_initialization
{
   public static void main(String args[])
   {
      double a, b;
      a = 3.0;
      b = 4.0;
      double c = Math.sqrt(a * a + b * b);
      System.out.println(c);
   }
}
```

a) 5.0
b) 25.0
c) 7.0
d) Compilation Error

**Answer : A**

Explanation: Variable c has been dynamically initialized to square root of a * a + b * b, during run time.

# Type Conversions, Promotions and Castings

**Question 74:** Which of these is necessary condition for automatic type conversion in Java?

a) The destination type is smaller than source type
b) The destination type is larger than source type
c) The destination type can be larger or smaller than source type
d) None of the mentioned

**Answer : B**

Explanation: None.

**Question 75:** What is the prototype of the default constructor of this class?

```
public class prototype { }
```

a) prototype( )
b) prototype(void)
c) public prototype(void)
d) public prototype( )

**Answer : D**

Explanation: None.

**Question 76:** What is the error in this code?

```
byte b = 50;
b = b * 50;
```

a) b cannot contain value 100, limited by its range
b) * operator has converted b * 50 into int, which cannot be converted to byte without casting
c) b cannot contain value 50
d) No error in this code

**Answer : B**

Explanation: While evaluating an expression containing int, bytes or shorts, the whole expression is converted to int then evaluated and the result is also of type int.

**Question 77:** If an expression contains double, int, float, long, then the whole expression will be promoted into which of these data types?

a) long
b) int
c) double
d) float

**Answer : C**

Explanation: If any operand is double the result of an expression is double.

**Question 78:** What is Truncation is Java?

a) Floating-point value assigned to an integer type
b) Integer value assigned to floating type
c) Floating-point value assigned to an Floating type
d) Integer value assigned to floating type

**Answer : A**

Explanation: None.

**Question 79:** What is the output of this program?

```
class char_increment
{
   public static void main(String args[])
   {
      char c1 = 'D';
      char c2 = 84;
      c2++;
      c1++;
      System.out.println(c1 + " "  + c2);
   }
}
```

a) E U
b) U E
c) V E
d) U F

**Answer : A**

Explanation: Operator ++ increments the value of character by 1. c1 and c2 are given values D and 84, when we use ++ operator their values increments by 1, c1 and c2 becomes E and U respectively.

**Question 80:** What is the output of this program?

```
class conversion
{
   public static void main(String args[])
   {
      double a = 295.04;
      int  b = 300;
      byte c = (byte) a;
      byte d = (byte) b;
      System.out.println(c + " "  + d);
   }
}
```

a) 38 43
b) 39 44
c) 295 300
d) 295.04 300

**Answer : B**

Explanation: Type casting a larger variable into a smaller variable results in modulo of larger variable by range of smaller variable. b contains 300 which is larger than byte's range i:e -128 to 127 hence d contains 300 modulo 256 i:e 44.

**Question 81:** What is the output of this program?

```
class A
{
   final public int calculate(int a, int b) { return 1; }
}
class B extends A
{
   public int calculate(int a, int b) { return 2; }
}
 public class output
 {
    public static void main(String args[])
    {
        B object = new B();
        System.out.print("b is " + b.calculate(0, 1));
    }
}
```

a) b is : 2
b) b is : 1
c) Compilation Error
d) An exception is thrown at runtime

**Answer : C**

Explanation: The code does not compile because the method calculate() in class A is final and so cannot be overridden by method of class b.

**Question 82:** What is the output of this program, if we run as "java main_arguments 1 2 3"?

```
class main_arguments
{
   public static void main(String [] args)
   {
      String [][] argument = new String[2][2];
      argument[0] = args;
      int x = argument[0].length;
      for (int y = 0; y < x; y++)
         System.out.print(" " + argument[0][y]);
   }
}
```

a) 1 1
b) 1 0
c) 1 0 3
d) 1 2 3

**Answer : D**

Explanation: In argument[0] = args;, the reference variable arg[0], which was referring to an array with two elements, is reassigned to an array (args) with three elements.

**Question 83:** What is the output of this program?

```
class c
{
   public void main( String[] args )
   {
      System.out.println( "Hello" + args[0] );
   }
}
```

a) Hello c
b) Hello
c) Hello world
d) Runtime Error

**Answer : D**

Explanation: A runtime error will occur owning to the main method of the code fragment not being declared static.

# Arrays

**Question 84:** Which of these operators is used to allocate memory to array variable in Java?

a) malloc
b) alloc
c) new
d) new malloc

**Answer : C**

Explanation: Operator new allocates a block of memory specified by the size of an array, and gives the reference of memory allocated to the array variable.

**Question 85:** Which of these is an incorrect array declaration?

a) int arr[] = new int[5];
b) int [] arr = new int[5].;
c) float arr[] = new float[5];
d) int arr[] = int [5] new;

**Answer : D**

Explanation: Operator new must be succeeded by array type and array size.

**Question 86:** What will this code print?

```
int arr[] = new int [5];
System.out.print(arr);
```

a) 0
b) value stored in arr[0].
c) 00000
d) Class name@ hashcode in hexadecimal form

**Answer : D**

Explanation: If we trying to print any reference variable internally, toString() will be called which is implemented to return the String in following form:
classname@hashcode in hexadecimal form



**Question 87:** Which of these is an incorrect Statement?

a) It is necessary to use new operator to initialize an array
b) Array can be initialized using comma separated expressions surrounded by curly braces
c) Array can be initialized when they are declared
d) None of the mentioned

**Answer : A**

Explanation: Array can be initialized using both new and comma separated expressions surrounded by curly braces example : int arr[5] = new int[5]; and int arr[] = { 0, 1, 2, 3, 4};

**Question 88:** What is the output of this program?

```
class array_output
{
   public static void main(String args[])
   {
      int array_variable [] = new int[10];
        for (int i = 0; i < 10; ++i)
       {
         array_variable[i] = i;
         System.out.print(array_variable[i] + " ");
         i++;
       }
   }
}
```

a) 0 2 4 6 8
b) 1 3 5 7 9
c) 0 1 2 3 4 5 6 7 8 9
d) 1 2 3 4 5 6 7 8 9 10

**Answer : A**

Explanation: When an array is declared using new operator then all of its elements are initialized to 0 automatically. for loop body is executed 5 times as whenever controls comes in the loop i value is incremented twice, first by i++ in body of loop then by ++i in increment condition of for loop.

**Question 89:** What is the output of this program?

```
class multidimention_array
{
   public static void main(String args[])
   {
      int arr[][] = new int[3][];
      arr[0] = new int[1];
      arr[1] = new int[2];
      arr[2] = new int[3];
       int sum = 0;
       for (int i = 0; i < 3; ++i)
             for (int j = 0; j < i + 1; ++j)
                   arr[i][j] = j + 1;
       for (int i = 0; i < 3; ++i)
             for (int j = 0; j < i + 1; ++j)
                   sum + = arr[i][j];
         System.out.print(sum);
   }
}
```

a) 11
b) 10
c) 13
d) 14

**Answer : B**

Explanation: arr[][] is a 2D array, array has been allotted memory in parts. 1st row contains 1 element, 2nd row contains 2 elements and 3rd row contains 3 elements. each element of array is given i + j value in loop. sum contains addition of all the elements of the array.

**Question 90:** What is the output of this program?

```
class evaluate
{
   public static void main(String args[])
      {
          int arr[] = new int[] {0 , 1, 2, 3, 4, 5, 6, 7, 8, 9};
          int n = 6;
          n = arr[arr[n] / 2];
          System.out.println(arr[n] / 2);
      }
}
```

a) 3
b) 0
c) 6
d) 1

**Answer : D**

Explanation: Array arr contains 10 elements. n contains 6 thus in next line n is given value 2 printing arr[2]/2 i:e 2/2 = 1.

**Question 91:** What is the output of this program?

```
class array_output
{
   public static void main(String args[])
   {
      char array_variable [] = new char[10];
      for (int i = 0; i < 10; ++i)
      {
            array_variable[i] = 'i';
            System.out.print(array_variable[i] + "");
      }
   }
}
```

a) 1 2 3 4 5 6 7 8 9 10
b) 0 1 2 3 4 5 6 7 8 9 10
c) i j k l m n o p q r
d) i i i i i i i i i i

**Answer : D**

Explanation: None.

**Question 92:** What is the output of this program?

```
class array_output
{
   public static void main(String args[])
   {
      int array_variable[][] = {{ 1, 2, 3}, { 4 , 5, 6}, { 7, 8, 9}};
      int sum = 0;
      for (int i = 0; i < 3; ++i)
            for (int j = 0; j <  3 ; ++j)
                  sum = sum + array_variable[i][j];
      System.out.print(sum / 5);
   }
}
```

a) 8
b) 9
c) 10
d) 11

**Answer : B**

Explanation: None.

# Data Structures-Arrays

**Question 93:** What is the type of variable 'b' and 'd' in the below snippet?

```
int a[], b;
int []c, d;
```

a) 'b' and 'd' are int
b) 'b' and 'd' are arrays of type int
c) 'b' is int variable; 'd' is int array
d) 'd' is int variable; 'b' is int array

**Answer : C**

Explanation: If [] is declared after variable it is applicable only to one variable. If [] is declared before variable it is applicable to all the variables.

**Question 94:** What is the output of below snippet?

```
Object[] names = new String[3];
names[0] = new Integer(0);
```

a) ArrayIndexOutOfBoundsException
b) ArrayStoreException
c) Compilation Error
d) Code runs successfully

**Answer : B**

Explanation: ArrayIndexOutOfBoundsException comes when code tries to access an invalid index for a given array. ArrayStoreException comes when you have stored an element of type other than the type of array.

**Question 95:** Generics does not work with?

a) Set
b) List
c) Tree
d) Array

**Answer : D**

Explanation: Generics gives the flexibility to strongly typecast collections. Generics is applicable to Set, List and Tree. It is not applicable to Array.

**Question 96:** How to sort an array?

a) Array.sort()
b) Arrays.sort()
c) Collection.sort()
d) System.sort()

**Answer : B**

Explanation: Arrays class contains various methods for manipulating arrays (such as sorting and searching). Array is not a valid class.

**Question 97:** How to copy contents of array?

a) System.arrayCopy()
b) Array.copy()
c) Arrays.copy()
d) Collection.copy()

**Answer : A**

Explanation: Arrays class contains various methods for manipulating arrays (such as sorting and searching). Array is not a valid class.

**Question 98:** Where is array stored in memory?

a) heap space
b) stack space
c) heap space and stack space
d) first generation memory

**Answer : A**

Explanation: Array is stored in heap space. Whenever an object is created, it's always stored in the Heap space and stack memory contains the reference to it.

**Question 99:** An array elements are always stored in _____ memory locations?

a) Sequential
b) Random
c) Sequential and Random
d) Binary search

**Answer : A**

Explanation: Array elements are stored in contiguous memory. Linked List is stored in random memory locations.

# Operators

**Question 100:** Which of the following can be operands of arithmetic operators?

a) Numeric
b) Boolean
c) Characters
d) Both Numeric & Characters

**Answer : D**

Explanation: The operand of arithmetic operators can be any of numeric or character type, But not boolean.

**Question 101:** Modulus operator, %, can be applied to which of these?

a) Integers
b) Floating – point numbers
c) Both Integers and floating – point numbers
d) None of the mentioned

**Answer : C**

Explanation: Modulus operator can be applied to both integers and floating point numbers.

**Question 102:** With x = 0, which of the following are legal lines of Java code for changing the value of x to 1?

1. x++;
2. x = x + 1;
3. x += 1;
4. x =+ 1;

a) 1, 2 & 3
b) 1 & 4
c) 1, 2, 3 & 4
d) 3 & 2

**Answer : C**

Explanation: Operator ++ increases value of variable by 1. x = x + 1 can also be written in shorthand form as x += 1. Also x =+ 1 will set the value of x to 1.

**Question 103:** Decrement operator, −−, decreases the value of variable by what number?

a) 1
b) 2
c) 3
d) 4

**Answer : A**

Explanation: None.

**Question 104:** Which of these statements are incorrect?

a) Assignment operators are more efficiently implemented by Java run-time system than their equivalent long forms
b) Assignment operators run faster than their equivalent long forms
c) Assignment operators can be used only with numeric and character data type
d) None of the mentioned

**Answer : D**

Explanation: None.

**Question 105:** What is the output of this program?

```
class increment
{
   public static void main(String args[])
   {
      double var1 = 1 + 5;
      double var2 = var1 / 4;
      int var3 = 1 + 5;
      int var4 = var3 / 4;
      System.out.print(var2 + " " + var4);
   }
}
```

a) 1 1
b) 0 1
c) 1.5 1
d) 1.5 1.0

**Answer : C**

Explanation: None.

**Question 106:** What is the output of this program?

```
class Modulus
{
   public static void main(String args[])
   {
      double a = 25.64;
      int  b = 25;
      a = a % 10;
      b = b % 10;
      System.out.println(a + " "  + b);
   }
}
```

a) 5.640000000000001 5
b) 5.640000000000001 5.0
c) 5 5
d) 5 5.640000000000001

**Answer : A**

Explanation: Modulus operator returns the remainder of a division operation on the operand. a = a % 10 returns 25.64 % 10 i:e 5.640000000000001. Similarly b = b % 10 returns 5.

**Question 107:** What is the output of this program?

```
class increment
{
    public static void main(String args[])
    {
        int g = 3;
        System.out.print(++g * 8);
    }
}
```

a) 25
b) 24
c) 32
d) 33

**Answer : C**

Explanation: Operator ++ has more preference than *, thus g becomes 4 and when multiplied by 8 gives 32.

**Question 108:** Can 8 byte long data type be automatically type cast to 4 byte float data type?

a a) True
b) False

**Answer : A**

Explanation: Both data types have different memory representation that's why 8-byte integral data type can be stored to 4-byte floating point data type.

**Question 109:** What is the output of this program?

```
class Output
{
    public static void main(String args[])
    {
        int a = 1, b=2, c, d;
        c = ++b;
        d = a++;
        c++;
        b++;
        ++a;
        System.out.println(a + " " + b + " " + c);
    }
}
```

a) 3 2 4
b) 3 2 3
c) 2 3 4
d) 3 4 4

**Answer : D**

Explanation: None.

**Question 110:** Which of these is not a bitwise operator?

a) &
b) &=
c) |=
d) <=

**Answer : D**

Explanation: <= is a relational operator.

**Question 111:** Which operator is used to invert all the digits in a binary representation of a number?

a) ~
b) <<<
c) >>>
d) ^

**Answer : A**

Explanation: Unary not operator, ~, inverts all of the bits of its operand in binary representation.

**Question 112:** What is the output of this program?

```
class bitwise_operator
{
   public static void main(String args[])
   {
      int var1 = 42;
      int var2 = ~var1;
      System.out.print(var1 + " " + var2);
   }
}
```

a) 42 42
b) 43 43
c) 42 -43
d) 42 43

**Answer : C**

Explanation: Unary not operator, ~, inverts all of the bits of its operand. 42 in binary is 00101010 in using ~ operator on var1 and assigning it to var2 we get inverted value of 42 i:e 11010101 which is -43 in decimal.

**Question 113:** What is the output of this program?

```
class bitwise_operator
{
   public static void main(String args[])
   {
       int a = 3;
       int b = 6;
        int c = a | b;
       int d = a & b;
       System.out.println(c + " "  + d);
   }
}
```

a) 7 2
b) 7 7
c) 7 5
d) 5 2

**Answer : A**

Explanation: And operator produces 1 bit if both operand are 1. Or operator produces 1 bit if any bit of the two operands in 1.

**Question 114:** What is the output of this program?

```
class leftshift_operator
{
   public static void main(String args[])
   {
       byte x = 64;
       int i;
       byte y;
       i = x << 2;
       y = (byte) (x << 2)
       System.out.print(i + " " + y);
   }
}
```

a) 0 64
b) 64 0
c) 0 256
d) 256 0

**Answer : D**

Explanation: None.

**Question 115:** What is the output of this program?

```
class rightshift_operator
{
    public static void main(String args[])
    {
        int x;
        x = 10;
        x = x >> 1;
        System.out.println(x);
    }
}
```

a) 10
b) 5
c) 2
d) 20

**Answer : B**

Explanation: Right shift operator, >>, devides the value by 2.

**Question 116:** What is the output of this program?

```
class Output
{
    public static void main(String args[])
    {
        int a = 1;
        int b = 2;
        int c = 3;
        a |= 4;
        b >>= 1;
        c <<= 1;
        a ^= c;
        System.out.println(a + " " + b + " " + c);
    }
}
```

a) 3 1 6
b) 2 2 3
c) 2 3 4
d) 3 3 6

**Answer : A**

Explanation: None.

**Question 117:** What is the output of relational operators?

a) Integer
b) Boolean
c) Characters
d) Double

**Answer : B**

Explanation: None.

**Question 118:** Which of these is returned by "greater than", "less than" and "equal to" operators?

a) Integers
b) Floating – point numbers
c) Boolean
d) None of the mentioned

**Answer : C**

Explanation: Explanation: All relational operators return a boolean value ie. true and false.

**Question 119:** Which of the following operators can operate on a boolean variable?

1. &&
2. ==
3. ?:
4. +=

a) 3 & 2
b) 1 & 4
c) 1, 2 & 4
d) 1, 2 & 3

**Answer : D**

Explanation: Operator Short circuit AND, &&, equal to, == , ternary if-then-else, ?:, are boolean logical operators. += is an arithmetic operator it can operate only on numeric values.

**Question 120:** Which of these operators can skip evaluating right hand operand?

a) !
b) |
c) &
d) &&

**Answer : D**

Explanation: Operator short circuit and, &&, and short circuit or, ||, skip evaluating right hand operand when output can be determined by left operand alone.

**Question 121:** Which of these statements is correct?

a) true and false are numeric values 1 and 0
b) true and false are numeric values 0 and 1
c) true is any non zero value and false is 0
d) true and false are non numeric values

**Answer : D**

Explanation: True and false are keywords, they are non numeric values which do not relate to zero or non zero numbers. true and false are boolean values.

**Question 122:** What is the output of this program?

```
class Relational_operator
{
   public static void main(String args[])
   {
      int var1 = 5, var2=6;
      System.out.print(var1 > var2);
   }
}
```

a) 1
b) 0
c) true
d) false

**Answer : D**

Explanation: Operator > returns a boolean value. 5 is not greater than 6 therefore false is returned.

**Question 123:** What is the output of this program?

```
class bool_operator
{
   public static void main(String args[])
   {
      boolean a = true;
      boolean b = !true;
      boolean c = a | b;
       boolean d = a & b;
      boolean e = d ? b : c;
      System.out.println(d + " " + e);
   }
}
```

a) false false
b) true ture
c) true false
d) false true

**Answer : D**

Explanation: Operator | returns true if any one operand is true, thus 'c = true | false' is true. Operator & returns a true if both of the operand is true thus d is false. Ternary operator ?: assigns left of ':' if condition is true and right hand of ':' if condition is false. d is false thus e = d ? b : c , assigns c to e , e contains true.

**Question 124:** What is the output of this program?

```
class ternary_operator
{
   public static void main(String args[])
   {
      int x = 3, z;
      int y = ~ x;
      z = x > y ? x : y;
      System.out.print(z);
   }
}
```

a) 0
b) 1
c) 3
d) -4

**Answer : C**

Explanation: None.

**Question 125:** What is the output of this program?

```
class Output
{
   public static void main(String args[])
   {
      int x = 10 , y = 1;
      if (x != 10 && x / 0 == 0)
         System.out.println(y);
      else
         System.out.println(++y);
   }
}
```

a) 1
b) 2
c) Runtime error owing to division by zero in if condition
d) Unpredictable behavior of program

**Answer : B**

Explanation: Operator short circuit and, &&, skips evaluating right hand operand if left hand operand is false thus division by zero in if condition does not give an error.

**Question 126:** What is the output of this program?

```
class Output
{
   public static void main(String args[])
   {
      boolean a = true;
      boolean b = false;
      boolean c = a ^ b;
      System.out.println(!c);
   }
}
```

a) 0
b) 1
c) false
d) true

**Answer : C**

Explanation: None.

**Question 127:** Which of these have highest precedence?

a) ()
b) ++
c) *
d) >>

**Answer : A**

Explanation: Order of precedence is (highest to lowest) a -> b -> c -> d.

**Question 128:** What should be expression1 evaluate to in using ternary operator as in this line?

```
expression1 ? expression2 : expression3
```

a) Integer
b) Floating – point numbers
c) Boolean
d) None of the mentioned

**Answer : C**

Explanation: The controlling condition of ternary operator must evaluate to boolean.

**Question 129:** What is the value stored in x in following lines of code?

```
int x, y, z;
x = 0;
y = 1;
x = y = z = 8;
```

a) 0
b) 1
c) 9
d) 8

**Answer : D**

Explanation: None.

**Question 130:** What is the order of precedence (highest to lowest) of following operators?

1. &
2. ^
3. ?:

a) 1 -> 2 -> 3
b) 2 -> 1 -> 3
c) 3 -> 2 -> 1
d) 2 -> 3 -> 1

**Answer : A**

Explanation: None.

**Question 131:** Which of these statements are incorrect?

a) Equal to operator has least precedence
b) Brackets () have highest precedence
c) Division operator, /, has higher precedence than multiplication operator
d) Addition operator, +, and subtraction operator have equal precedence

**Answer : C**

Explanation: Division operator, /, has equal precedence as of multiplication operator. In expression involving multiplication and division evaluation of expression will begin from the right side when no brackets are used.

**Question 132:** What is the output of this program?

```
class operators
{
   public static void main(String args[])
   {
      int var1 = 5;
      int var2 = 6;
      int var3;
      var3 = ++ var2 * var1 / var2 + var2;
      System.out.print(var3);
   }
}
```

a) 10
b) 11
c) 12
d) 56

**Answer : C**

Explanation: Operator ++ has the highest precedence than /, * and +. var2 is incremented to 7 and then used in expression, var3 = 7 * 5 / 7 + 7, gives 12.

**Question 133:** What is the output of this program?

```
class operators
{
   public static void main(String args[])
   {
      int x = 8;
      System.out.println(++x * 3 + " " + x);
   }
}
```

a) 24 8
b) 24 9
c) 27 8
d) 27 9

**Answer : D**

Explanation: Operator ++ has higher precedence than multiplication operator, *, x is incremented to 9 than multiplied with 3 giving 27.

**Question 134:** What is the output of this program?

```
class Output
{
    public static void main(String args[])
    {
        int x=y=z=20;
    }
}
```

a) compile and runs fine
b) 20
c) run time error
d) compile time error

**Answer : D**

Explanation: None.

**Question 135:** Which of these lines of code will give better performance?

1. a | 4 + c >> b & 7;
2. a | ((( 4 * c ) >> b ) & 7 ))

a) 1 will give better performance as it has no parentheses
b) 2 will give better performance as it has parentheses
c) Both 1 & 2 will give equal performance
d) Dependent on the computer system

**Answer : C**

Explanation: Parentheses do not degrade the performance of the program. Adding parentheses to reduce ambiguity does not negatively affect your system.

**Question 136:** What is the output of this program?

```
class Output
{
    public static void main(String args[])
    {
        int a,b,c,d;
        a=b=c=d=20;
        a+=b-=c*=d/=20;
        System.out.println(a+" "+b+" "+c+" "+d);
    }
}
```

a) compile time error
b) runtime error
c) a=20 b=0 c=20 d=1
d) none of the mentioned

**Answer : C**

Explanation: Expression will evaluate from right to left.

# Control Statements

**Question 137:** Which of these selection statements test only for equality?

a) if
b) switch
c) if & switch
d) none of the mentioned

**Answer : B**

Explanation: Switch statements checks for equality between the controlling variable and its constant cases.

**Question 138:** Which of these are selection statements in Java?

a) if()
b) for()
c) continue
d) break

**Answer : A**

Explanation: Continue and break are jump statements, and for is a looping statement.

**Question 139:** Which of the following loops will execute the body of loop even when condition controlling the loop is initially false?

a) do-while
b) while
c) for
d) none of the mentioned

**Answer : A**

Explanation: None.

**Question 140:** Which of these jump statements can skip processing the remainder of the code in its body for a particular iteration?

a) break
b) return
c) exit
d) continue

**Answer : D**

Explanation: None.

**Question 141:** Which of this statement is incorrect?

a) switch statement is more efficient than a set of nested ifs
b) two case constants in the same switch can have identical values
c) switch statement can only test for equality, whereas if statement can evaluate any type of boolean expression
d) it is possible to create a nested switch statements

**Answer : B**

Explanation: No two case constants in the same switch can have identical values.

**Question 142:** What is the output of this program?

```
class selection_statements
{
   public static void main(String args[])
   {
      int var1 = 5;
      int var2 = 6;
      if ((var2 = 1) == var1)
         System.out.print(var2);
      else
         System.out.print(++var2);
   }
}
```

a) 1
b) 2
c) 3
d) 4

**Answer : B**

Explanation: var2 is initialised to 1. The conditional statement returns false and the else part gets executed.



**Question 143:** What is the output of this program?

```
class comma_operator
{
   public static void main(String args[])
   {
       int sum = 0;
       for (int i = 0, j = 0; i < 5 & j < 5; ++i, j = i + 1)
          sum += i;
         System.out.println(sum);
   }
}
```

a) 5
b) 6
c) 14
d) compilation error

**Answer : B**

Explanation: Using comma operator, we can include more than one statement in the initialization and iteration portion of the for loop. Therefore both ++i and j = i + 1 is executed i gets the value – 0,1,2,3,4 & j gets the values -0,1,2,3,4,5.

**Question 144:** What is the output of this program?

```
class jump_statments
{
   public static void main(String args[])
   {
      int x = 2;
      int y = 0;
      for ( ; y < 10; ++y)
      {
         if (y % x == 0)
            continue;
         else if (y == 8)
            break;
         else
           System.out.print(y + " ");
      }
   }
}
```

a) 1 3 5 7
b) 2 4 6 8
c) 1 3 5 7 9
d) 1 2 3 4 5 6 7 8 9

**Answer : C**

Explanation: Whenever y is divisible by x remainder body of loop is skipped by continue statement, therefore if condition y == 8 is never true as when y is 8, remainder body of loop is skipped by continue statements of first if. Control comes to print statement only in cases when y is odd.

**Question 145:** What is the output of this program?

```
class Output
{
    public static void main(String args[])
    {
       final int a=10,b=20;
       while(a<b)
       {
           System.out.println("Hello");
       }
       System.out.println("World");
    }
}
```

a) Hello
b) run time error
c) Hello world
d) compile time error

**Answer : D**

Explanation: Every final variable is compile time constant.

**Question 146:** What is the output of this program?

```
class Output
{
   public static void main(String args[])
   {
      int a = 5, b=10;
      first:
      {
        second:
        {
          third:
          {
             if (a ==  b >> 1)
                break second;
          }
          System.out.println(a);
        }
        System.out.println(b);
      }
   }
}
```

a) 5 10
b) 10 5
c) 5
d) 10

**Answer : D**

Explanation: b >> 1 in if returns 5 which is equal to a i:e 5, therefore body of if is executed and block second is exited. Control goes to end of the block second executing the last print statement, printing 10.

**Question 147:** What would be the output of the following codesnippet if variable a=10?

```
if(a<=0)
{
  if(a==0)
  {
    System.out.println("1 ");
  }
  else
  {
    System.out.println("2 ");
  }
}
System.out.println("3 ");
```

a) 1 2
b) 2 3
c) 1 3
d) 3

**Answer : D**

Explanation: Since the first if condition is not met, control would not go inside if statement and hence only statement after the entire if block will be executed.



**Question 148:** The while loop repeats a set of code while the condition is not met?

a) True
b) False

**Answer : B**

Explanation: While loop repeats a set of code only until the condition is met.

**Question 149:** What is true about a break?

a) Break stops the execution of entire program
b) Break halts the execution and forces the control out of the loop
c) Break forces the control out of the loop and starts the execution of next iteration
d) Break halts the execution of the loop for certain time frame

**Answer : B**

Explanation: Break halts the execution and forces the control out of the loop.

**Question 150:** What is true about do statement?

a) do statement executes the code of a loop at least once
b) do statement does not get execute if condition is not matched in the first iteration
c) do statement checks the condition at the beginning of the loop
d) do statement executes the code more than once always

**Answer : A**

Explanation: Do statement checks the condition at the end of the loop. Hence, code gets executed at least once.

**Question 151:** Which of the following is used with the switch statement?

a) Continue
b) Exit
c) break
d) do

**Answer : C**

Explanation: Break is used with a switch statement to shift control out of switch.

**Question 152:** What is the valid data type for variable "a" to print "Hello World"?

```
switch(a)
{
   System.out.println("Hello World");
}
```

a) int and float
b) byte and short
c) char and long
d) byte and char

**Answer : D**

Explanation: The switch condition would only meet if variable "a" is of type byte or char.

**Question 153:** Which of the following is not a decision making statement?

a) if
b) if-else
c) switch
d) do-while

**Answer : D**

Explanation: do-while is an iteration statement. Others are decision making statements.

**Question 154:** Which of the following is not a valid jump statement?

a) break
b) goto
c) continue
d) return

**Answer : B**

Explanation: break, continue and return transfer control to another part of the program and returns back to caller after execution. However, goto is marked as not used in Java.

**Question 155:** From where break statement causes an exit?

a) Only from innermost loop
b) Terminates a program
c) Only from innermost switch
d) From innermost loops or switches

**Answer : D**

Explanation: The break statement causes an exit from innermost loop or switch.

**Question 156:** Which of the following is not a valid flow control statement?

a) exit()
b) break
c) continue
d) return

**Answer : A**

Explanation: exit() is not a flow control statement in Java. exit() terminates the currently running JVM.

# Concepts of OOPs

**Question 157:** Which of the following is not OOPS concept in Java?

a) Inheritance
b) Encapsulation
c) Polymorphism
d) Compilation

**Answer : D**

Explanation: There are 4 OOPS concepts in Java. Inheritance, Encapsulation, Polymorphism and Abstraction.

**Question 158:** Which of the following is a type of polymorphism in Java?

a) Compile time polymorphism
b) Execution time polymorphism
c) Multiple polymorphism
d) Multilevel polymorphism

**Answer : A**

Explanation: There are two types of polymorphism in Java: Compile time polymorphism (overloading) and runtime polymorphism (overriding).

**Question 159:** When does method overloading is determined?

a) At run time
b) At compile time
c) At coding time
d) At execution time

**Answer : B**

Explanation: Overloading is determined at compile time. Hence, it is also known as compile time polymorphism.

**Question 160:** When Overloading does not occur?

a) More than one method with same name but different method signature and different number or type of parameters
b) More than one method with same name, same signature but different number of signature
c) More than one method with same name, same signature, same number of parameters but different type
d) More than one method with same name, same number of parameters and type but different signature

**Answer : D**

Explanation: Overloading occurs when more than one method with same name but different constructor and also when same signature but different number of parameters and/or parameter type.

**Question 161:** Which concept of Java is a way of converting real world objects in terms of class?

a) Polymorphism
b) Encapsulation
c) Abstraction
d) Inheritance

**Answer : C**

Explanation: Abstraction is the concept of defining real world objects in terms of classes or interfaces.

**Question 162:** Which concept of Java is achieved by combining methods and attribute into a class?

a) Encapsulation
b) Inheritance
c) Polymorphism
d) Abstraction

**Answer : A**

Explanation: Encapsulation is implemented by combining methods and attribute into a class. The class acts like a container of encapsulating properties.

**Question 163:** What is it called where child object gets killed if parent object is killed?

a) Aggregation
b) Composition
c) Encapsulation
d) Association

**Answer : A**

Explanation: Composition occurs when child object gets killed if parent object gets killed.

**Question 164:** Method overriding is combination of inheritance and polymorphism?

a) True
b) false

**Answer : A**

Explanation: In order for method overriding, method with same signature in both superclass and subclass is required with same signature. That satisfies both concepts inheritance and polymorphism.

# JDK-JRE-JIT-JVM

**Question 165:** Which component is used to compile, debug and execute java program?

a) JVM
b) JDK
c) JIT
d) JRE

**Answer : B**

Explanation: JDK is a core component of Java Environment and provides all the tools, executables and binaries required to compile, debug and execute a Java Program.

**Question 166:** Which component is responsible for converting bytecode into machine specific code?

a) JVM
b) JDK
c) JIT
d) JRE

**Answer : A**

Explanation: JVM is responsible to converting bytecode to the machine specific code. JVM is also platform dependent and provides core java functions like garbage collection, memory management, security etc.

**Question 167:** Which component is responsible to run java program?

a) JVM
b) JDK
c) JIT
d) JRE

**Answer : D**

Explanation: JRE is the implementation of JVM, it provides platform to execute java programs.

**Question 168:** Which component is responsible to optimize bytecode to machine code?

a) JVM
b) JDK
c) JIT
d) JRE

**Answer : C**

Explanation: JIT optimizes bytecode to machine specific language code by compiling similar bytecodes at the same time. This reduces overall time taken for compilation of bytecode to machine specific language.

**Question 169:** Which statement is true about java?

a) Platform independent programming language
b) Platform dependent programming language
c) Code dependent programming language
d) Sequence dependent programming language

**Answer : A**

Explanation: Java is called 'Platform Independent Language' as it primarily works on the principle of 'compile once, run everywhere'.

**Question 170:** Which of the below is invalid identifier with the main method?

a) public
b) static
c) private
d) final

**Answer : C**

Explanation: main method cannot be private as it is invoked by external method. Other identifier are valid with main method.

**Question 171:** What is the extension of java code files?

a) .class
b) .java
c) .txt
d) .js

**Answer : B**

Explanation: Java files have .java extension.

**Question 172:** What is the extension of compiled java classes?

a) .class
b) .java
c) .txt
d) .js

**Answer : A**

Explanation: The compiled java files have .class extension.

**Question 173:** How can we identify whether a compilation unit is class or interface from a .class file?

a) Java source file header
b) Extension of compilation unit
c) We cannot differentiate between class and interface
d) The class or interface name should be postfixed with unit type

**Answer : A**

Explanation: The Java source file contains a header that declares the type of class or interface, its visibility with respect to other classes, its name and any superclass it may extend, or interface it implements.

**Question 174:** What is use of interpreter?

a) They convert bytecode to machine language code
b) They read high level code and execute them
c) They are intermediated between JIT and JVM
d) It is a synonym for JIT

**Answer : B**

Explanation: Explanation: Interpreters read high level language (interprets it) and execute the program. Interpreters are normally not passing through byte-code and jit compilation.

# Class Fundamentals & Declaring objects

**Question 175:** What is the stored in the object obj in following lines of code?
box obj;

a) Memory address of allocated memory of object
b) NULL
c) Any arbitrary pointer
d) Garbage

**Answer : B**

Explanation: Memory is allocated to an object using new operator. box obj; just declares a reference to object, no memory is allocated to it hence it points to NULL.

**Question 176:** Which of these keywords is used to make a class?

a) class
b) struct
c) int
d) none of the mentioned

**Answer : A**

Explanation: None.

**Question 177:** Which of the following is a valid declaration of an object of class Box?

a) Box obj = new Box();
b) Box obj = new Box;
c) obj = new Box();
d) new Box obj;

**Answer : A**

Explanation: None.

**Question 178:** Which of these operators is used to allocate memory for an object?

a) malloc
b) alloc
c) new
d) give

**Answer : C**

Explanation: Operator new dynamically allocates memory for an object and returns a reference to it. This reference is address in memory of the object allocated by new.

**Question 179:** Which of these statement is incorrect?

a) Every class must contain a main() method
b) Applets do not require a main() method at all
c) There can be only one main() method in a program
d) main() method must be made public

**Answer : A**

E Explanation: Every class does not need to have a main() method, there can be only one main() method which is made public.

**Question 180:** What is the output of this program?

```
class main_class
{
   public static void main(String args[])
   {
      int x = 9;
      if (x == 9)
      {
         int x = 8;
         System.out.println(x);
      }
   }
}
```

a) 9
b) 8
c) Compilation error
d) Runtime error

**Answer : C**

Explanation: Two variables with the same name can't be created in a class.



**Question 181:** Which of the following statements is correct?

a) Public method is accessible to all other classes in the hierarchy
b) Public method is accessible only to subclasses of its parent class
c) Public method can only be called by object of its class
d) Public method can be accessed by calling object of the public class

**Answer : A**

Explanation: None.

**Question 182:** What is the output of this program?

```
class box
{
   int width;
   int height;
   int length;
}
class mainclass
{
   public static void main(String args[])
   {
      box obj = new box();
      obj.width = 10;
      obj.height = 2;
      obj.length = 10;
      int y = obj.width * obj.height * obj.length;
      System.out.print(y);
   }
}
```

a) 12
b) 200
c) 400
d) 100

**Answer : B**

Explanation: None.

**Question 183:** What is the output of this program?

```
class box
{
   int width;
   int height;
   int length;
}
class mainclass
{
   public static void main(String args[])
   {
      box obj1 = new box();
      box obj2 = new box();
      obj1.height = 1;
      obj1.length = 2;
      obj1.width = 1;
      obj2 = obj1;
      System.out.println(obj2.height);
   }
}
```

a) 1
b) 2
c) Runtime error
d) Garbage value

**Answer : A**

Explanation: When we assign an object to another object of same type, all the elements of right side object gets copied to object on left side of equal to, =, operator.

**Question 184:** What is the output of this program?

```
class box
{
    int width;
    int height;
    int length;
}
 class mainclass
 {
    public static void main(String args[])
    {
       box obj = new box();
       System.out.println(obj);
    }
 }
```

a) 0
b) 1
c) Runtime error
d) classname@hashcode in hexadecimal form

**Answer : D**

Explanation: When we print object internally toString() will be called to return string into this format classname@hashcode in hexadecimal form.

# Introduction to Methods

**Question 185:** What is the return type of a method that does not return any value?

a) int
b) float
c) void
d) double

**Answer : C**

Explanation: Return type of an method must be made void if it is not returning any value.

**Question 186:** What is the process of defining more than one method in a class differentiated by method signature?

a) Function overriding
b) Function overloading
c) Function doubling
d) None of the mentioned

**Answer : B**

Explanation: Function overloading is a process of defining more than one method in a class with same name differentiated by function signature i:e return type or parameters type and number. Example – int volume(int length, int width) & int volume(int length , int width , int height) can be used to calculate volume.



**Question 187:** Which of the following is a method having same name as that of it's class?

a) finalize
b) delete
c) class
d) constructor

**Answer : D**

Explanation: A constructor is a method that initializes an object immediately upon creation. It has the same name as that of class in which it resides.

**Question 188:** Which method can be defined only once in a program?

a) main method
b) finalize method
c) static method
d) private method

**Answer : A**

Explanation: main() method can be defined only once in a program. Program execution begins from the main() method by java runtime system.

**Question 189:** Which of this statement is incorrect?

a) All object of a class are allotted memory for the all the variables defined in the class
b) If a function is defined public it can be accessed by object of other class by inheritation
c) main() method must be made public
d) All object of a class are allotted memory for the methods defined in the class

**Answer : D**

Explanation: All object of class share a single copy of methods defined in a class, Methods are allotted memory only once. All the objects of the class have access to methods of that class are allotted memory only for the variables not for the methods.

**Question 190:** What is the output of this program?

```
class box
{
   int width, height, length, volume;
   void volume(int height, int length, int width)
   {
       volume = width*height*length;
   }
}
class Prameterized_method
{
   public static void main(String args[])
   {
      box obj = new box();
      obj.height = 1;
      obj.length = 2;
      obj.width = 5;
      obj.volume(3,2,1);
      System.out.println(obj.volume);
   }
}
```

a) 0
b) 1
c) 6
d) 25

**Answer : C**

Explanation: None.

**Question 191:** What is the output of this program?

```
class equality
{
   int x;
   int y;
   boolean isequal()
   {
      return(x == y);
   }
}
class Output
{
   public static void main(String args[])
   {
      equality obj = new equality();
      obj.x = 5;
      obj.y = 5;
      System.out.println(obj.isequal());
   }
}
```

a) false
b) true
c) 0
d) 1

**Answer : B**

Explanation: None.

**Question 192:** What is the output of this program?

```
class box
{
   int width;
   int height;
   int length;
   int volume;
   void volume()
   {
       volume = width*height*length;
   }
}
```

```
class Output
{
   public static void main(String args[])
   {
      box obj = new box();
      obj.height = 1;
      obj.length = 5;
      obj.width = 5;
      obj.volume();
      System.out.println(obj.volume);
   }
}
```

a) 0
b) 1
c) 25
d) 26

**Answer : C**

Explanation: None.

**Question 193:** What is the output of this program?

```
class area
{
   int width;
   int length;
   int volume;
   area()
   {
      width=5;
      length=6;
   }
   void volume()
   {
       volume = width*length*height;
   }
}
class cons_method
{
   public static void main(String args[])
   {
      area obj = new area();
      obj.volume();
      System.out.println(obj.volume);
   }
}
```

a) 0
b) 1
c) 30
d) error

**Answer : D**

Explanation: Variable height is not defined.

# Constructors & Garbage Collection

**Question 194:** What is the return type of Constructors?

a) int
b) float
c) void
d) none of the mentioned

**Answer : D**

Explanation: Constructors does not have any return type, not even void.

**Question 195:** Which keyword is used by the method to refer to the object that invoked it?

a) import
b) catch
c) abstract
d) this

**Answer : D**

Explanation: this keyword can be used inside any method to refer to the current object. this is always a reference to the object on which the method was invoked.

**Question 196:** Which of the following is a method having same name as that of its class?

a) finalize
b) delete
c) class
d) constructor

**Answer : D**

Explanation: A constructor is a method that initializes an object immediately upon creation. It has the same name as that of class in which it resides.

**Question 197:** Which operator is used by Java run time implementations to free the memory of an object when it is no longer needed?

a) delete
b) free
c) new
d) none of the mentioned

**Answer : D**

Explanation: Java handles deallocation of memory automatically, we do not need to explicitly delete an element. Garbage collection only occurs during execution of the program. When no references to the object exist, that object is assumed to be no longer needed, and the memory occupied by the object can be reclaimed.

**Question 198:** Which function is used to perform some action when the object is to be destroyed?

a) finalize()
b) delete()
c) main()
d) none of the mentioned

**Answer : A**

Explanation: None.

**Question 199:** What is the output of this program?

```
class box
{
   int width;
   int height;
   int length;
   int volume;
   box()
   {
      width = 5;
      height = 5;
      length = 6;
   }
   void volume()
   {
       volume = width*height*length;
   }
}
class constructor_output
{
   public static void main(String args[])
   {
      box obj = new box();
      obj.volume();
      System.out.println(obj.volume);
   }
}
```

a) 100
b) 150
c) 200
d) 250

**Answer : B**

Explanation: None.

**Question 200:** What is the output of this program?

```
class San
{
    San()throws IOException
    { }
}
class Foundry extends San
{
    Foundry()
    { }
    public static void main(String[]args)
    {

    }
}
```

a) compile time error
b) run time error
c) compile and runs fine
d) unreported exception java.io.IOException in default constructor

**Answer : A**

Explanation: If parent class constructor throws any checked exception, compulsory child class constructor should throw the same checked exception as its parent, otherwise code won't compile.

**Question 201:** What is the output of this program?

```
class box
{
   int width;
   int height;
   int length;
   int volume;
   void finalize()
   {
      volume = width*height*length;
      System.out.println(volume);
   }
   protected void volume()
   {
      volume = width*height*length;
      System.out.println(volume);
   }
}
```

```
class Output
{
   public static void main(String args[])
   {
      box obj = new box();
      obj.width=5;
      obj.height=5;
      obj.length=6;
      obj.volume();
   }
}
```

a) 150
b) 200
c) Run time error
d) Compilation error

**Answer : A**

Explanation: None.

**Question 202:** Which of the following statements are incorrect?

a) default constructor is called at the time of object declaration
b) Constructor can be parameterized
c) finalize() method is called when object goes out of scope and is no longer needed
d) finalize() method must be declared protected

**Answer : C**

Explanation: finalize() method is called just prior to garbage collection. it is not called when object goes out of scope.

**Question 203:** What should be expression1 evaluate to in using ternary operator as in this line?
expression1 ? expression2 : expression3

a) Integer
b) Floating – point numbers
c) Boolean
d) None of the mentioned

**Answer : C**

Explanation: The controlling condition of ternary operator must evaluate to boolean.

**Question 204:** What is the output of this program?

```
class area
{
   int width;
   int length;
   int area;
   void area(int width, int length)
   {
      this.width = width;
      this.length = length;
   }
}
class Output
{
   public static void main(String args[])
   {
      area obj = new area();
      obj.area(5 , 6);
      System.out.println(obj.length + " " + obj.width);
   }
}
```

a) 0 0
b) 5 6
c) 6 5
d) 5 5

**Answer : C**

Explanation: this keyword can be used inside any method to refer to the current object. this is always a reference to the object on which the method was invoked.

# Constructor

**Question 205:** What is true about private constructor?

a) Private constructor ensures only one instance of a class exist at any point of time
b) Private constructor ensures multiple instances of a class exist at any point of time
c) Private constructor eases the instantiation of a class
d) Private constructor allows creating objects in other classes

**Answer : A**

Explanation: Object of private constructor can only be created within class. Private constructor is used in singleton pattern.

**Question 206:** What is true about Class.getInstance()?

a) Class.getInstance calls the constructor
b) Class.getInstance is same as new operator
c) Class.getInstance needs to have matching constructor
d) Class.getInstance creates object if class does not have any constructor

**Answer : D**

Explanation: Class class provides list of methods for use like getInstance().

**Question 207:** What is true about constructor?

a) It can contain return type
b) It can take any number of parameters
c) It can have any non access modifiers
d) Constructor cannot throw an exception

**Answer : B**

Explanation: Constructor returns a new object with variables defined as in the class. Instance variables are newly created and only one copy of static variables are created.

**Question 208:** What would be the behaviour if one parameterized constructor is explicitly defined?

a) Compilation error
b) Compilation succeeds
c) Runtime error
d) Compilation succeeds but at the time of creating object using default constructor, it throws compilation error

**Answer : D**

Explanation: The class compiles successfully. But the object creation of that class gives a compilation error.

**Question 209:** What would be behaviour if the constructor has a return type?

a) Compilation error
b) Runtime error
c) Compilation and runs successfully
d) Only String return type is allowed

**Answer : A**

Explanation: The constructor cannot have a return type. It should create and return new object. Hence it would give compilation error.

# Heap and Garbage Collection

**Question 210:** Which of the following has the highest memory requirement?

a) Heap
b) Stack
c) JVM
d) Class

**Answer : C**

Explanation: JVM is the super set which contains heap, stack, objects, pointers, etc.

**Question 211:** Which of the following is a garbage collection technique?

a) Cleanup model
b) Mark and sweep model
c) Space management model
d) Sweep model

**Answer : B**

Explanation: A mark and sweep garbage collection consists of two phases, the mark phase and the sweep phase. I mark phase all the objects reachable by java threads, native handles and other root sources are marked alive and others are garbage. In sweep phase, the heap is traversed to find gaps between live objects and the gaps are marked free list used for allocating memory to new objects.

**Question 212:** Which exception is thrown when java is out of memory?

a) MemoryFullException
b) MemoryOutOfBoundsException
c) OutOfMemoryError
d) MemoryError

**Answer : C**

Explanation: The Xms flag has no default value, and Xmx typically has a default value of 256MB. A common use for these flags is when you encounter a java.lang.OutOfMemoryError.

**Question 213:** What happens to the thread when garbage collection kicks off?

a) The thread continues its operation
b) Garbage collection cannot happen until the thread is running
c) The thread is paused while garbage collection runs
d) The thread and garbage collection do not interfere with each other

**Answer : C**

Explanation: The thread is paused when garbage collection runs which slows the application performance.

**Question 214:** Garbage Collection can be controlled by a program?

a) True
b) False

**Answer : B**

Explanation: Garbage Collection cannot be controlled by a program.

# Overloading Methods & Argument Passing

**Question 215:** What is the process of defining two or more methods within same class that have same name but different parameters declaration?

a) method overloading
b) method overriding
c) method hiding
d) none of the mentioned

**Answer : A**

Explanation: Two or more methods can have same name as long as their parameters declaration is different, the methods are said to be overloaded and process is called method overloading. Method overloading is a way by which Java implements polymorphism.

**Question 216:** Which of these can be overloaded?

a) Methods
b) Constructors
c) All of the mentioned
d) None of the mentioned

**Answer : C**

Explanation: None.

**Question 217:** Which of these is correct about passing an argument by call-by-value process?

a) Copy of argument is made into the formal parameter of the subroutine
b) Reference to original argument is passed to formal parameter of the subroutine
c) Copy of argument is made into the formal parameter of the subroutine and changes made on parameters of subroutine have effect on original argument
d) Reference to original argument is passed to formal parameter of the subroutine and changes made on parameters of subroutine have effect on original argument

**Answer : A**

Explanation: When we pass an argument by call-by-value a copy of argument is made into the formal parameter of the subroutine and changes made on parameters of subroutine have no effect on original argument, they remain the same.

**Question 218:** What is the process of defining a method in terms of itself, that is a method that calls itself?

a) Polymorphism
b) Abstraction
c) Encapsulation
d) Recursion

**Answer : D**

Explanation: None.

**Question 219:** What is the output of the following code?

```
class San
{
   public void m1 (int i,float f)
   {
        System.out.println(" int float method");
   }
   public void m1(float f,int i);
   {
       System.out.println("float int method");
   }
  public static void main(String[]args)
  {
       San s=new San();
      s.m1(20,20);
  }
}
```

a) int float method
b) float int method
c) compile time error
d) run time error

**Answer : C**

Explanation: While resolving overloaded method, compiler automatically promotes if exact match is not found. But in this case, which one to promote is an ambiguity.

**Question 220:** What is the output of this program?

```
class overload
{
   int x;
   int y;
   void add(int a)
   {
      x =  a + 1;
   }
   void add(int a, int b)
   {
      x =  a + 2;
   }
}
```

```
class Overload_methods
{
   public static void main(String args[])
   {
      overload obj = new overload();
      int a = 0;
      obj.add(6);
      System.out.println(obj.x);
   }
}
```

a) 5
b) 6
c) 7
d) 8

**Answer : C**

Explanation: None.

**Question 221:** What is the output of this program?

```
class overload
{
   int x;
   int y;
   void add(int a)
   {
      x =  a + 1;
   }
   void add(int a, int b)
   {
      x =  a + 2;
   }
}
class Overload_methods
{
   public static void main(String args[])
   {
      overload obj = new overload();
      int a = 0;
      obj.add(6, 7);
      System.out.println(obj.x);
   }
}
```

a) 6
b) 7
c) 8
d) 9

**Answer : C**

Explanation: None.

**Question 222:** What is the output of this program?

```
class overload
{
    int x;
    double y;
    void add(int a , int b)
    {
       x = a + b;
    }
   void add(double c , double d)
   {
       y = c + d;
   }
   overload()
   {
       this.x = 0;
       this.y = 0;
   }
}
class Overload_methods
{
   public static void main(String args[])
   {
       overload obj = new overload();
       int a = 2;
       double b = 3.2;
       obj.add(a, a);
       obj.add(b, b);
       System.out.println(obj.x + " " + obj.y);
   }
}
```

a) 6 6
b) 6.4 6.4
c) 6.4 6
d) 4 6.4

**Answer : D**

Explanation: For obj.add(a,a); ,the function in line number 4 gets executed and value of x is 4. For the next function call, the function in line number 7 gets executed and value of y is 6.4.

**Question 223:** What is the output of this program?

```
class test
{
   int a;
   int b;
   void meth(int i , int j)
   {
      i *= 2;
      j /= 2;
   }
}
class Output
{
   public static void main(String args[])
   {
      test obj = new test();
      int a = 10;
      int b = 20;
      obj.meth(a , b);
      System.out.println(a + " " + b);
   }
}
```

a) 10 20
b) 20 10
c) 20 40
d) 40 20



**Answer : A**

Explanation: Variables a & b are passed by value, copy of their values are made on formal parameters of function meth() that is i & j. Therefore changes done on i & j are not reflected back on original arguments. a & b remain 10 & 20 respectively.

**Question 224:** What is the output of this program?

```
class test
{
   int a;
   int b;
   test(int i, int j)
   {
      a = i;
      b = j;
   }
   void meth(test o)
   {
      o.a *= 2;
      O.b /= 2;
   }
}
class Output
{
   public static void main(String args[])
   {
      test obj = new test(10 , 20);
      obj.meth(obj);
      System.out.println(obj.a + " " + obj.b);
   }
  }
```

a) 10 20
b) 20 10
c) 20 40
d) 40 20

**Answer : B**

Explanation: Class objects are always passed by reference, therefore changes done are reflected back on original arguments. obj.meth(obj) sends object obj as parameter whose variables a & b are multiplied and divided by 2 respectively by meth() function of class test. a & b becomes 20 & 10 respectively.

# Access Control

**Question 225:** Which of these access specifiers must be used for main() method?

a) private
b) public
c) protected
d) none of the mentioned

**Answer : B**

Explanation: main() method must be specified public as it called by Java run time system, outside of the program. If no access specifier is used then by default member is public within its own package & cannot be accessed by Java run time system.

**Question 226:** Which of these is used to access a member of class before object of that class is created?

a) public
b) private
c) static
d) protected

**Answer : C**

Explanation: None.

**Question 227:** What is the process by which we can control what parts of a program can access the members of a class?

a) Polymorphism
b) Abstraction
c) Encapsulation
d) Recursion

**Answer : C**

Explanation: None.

**Question 228:** Which of the following statements are incorrect?

a) public members of class can be accessed by any code in the program
b) private members of class can only be accessed by other members of the class
c) private members of class can be inherited by a subclass, and become protected members in subclass
d) protected members of a class can be inherited by a subclass, and become private members of the subclass

**Answer : C**

Explanation: private members of a class can not be inherited by a subclass.

**Question 229:** What is the output of this program?

```
class access
{
   public int x;
   private int y;
   void cal(int a, int b)
   {
      x =  a + 1;
      y =  b;
   }
}
class access_specifier
{
   public static void main(String args[])
   {
      access obj = new access();
      obj.cal(2, 3);
      System.out.println(obj.x + " " + obj.y);
   }
}
```

a) 3 3
b) 2 3
c) Runtime Error
d) Compilation Error

**Answer : C**

Explanation: None.

**Question 230:** What is the output of this program?

```
class access
{
   public int x;
   private int y;
   void cal(int a, int b)
   {
      x =  a + 1;
      y =  b;
   }
   void print()
   {
      system.out.println(" " + y);
   }
}
```

```
class access_specifier
{
   public static void main(String args[])
   {
      access obj = new access();
      obj.cal(2, 3);
      System.out.println(obj.x);
      obj.print();
   }
}
```

a) 2 3
b) 3 3
c) Runtime Error
d) Compilation Error

**Answer : B**

Explanation: None.

**Question 231:** What is the output of this program?

```
class static_out
{
   static int x;
   static int y;
   void add(int a, int b)
   {
      x = a + b;
      y = x + b;
   }
}
class static_use
{
   public static void main(String args[])
   {
      static_out obj1 = new static_out();
      static_out obj2 = new static_out();
      int a = 2;
      obj1.add(a, a + 1);
      obj2.add(5, a);
      System.out.println(obj1.x + " " + obj2.y);
   }
}
```

a) 7 7.4
b) 6 6.4
c) 7 9
d) 9 7

**Answer : B**

Explanation: None.

**Question 232:** Which one of the following is not an access modifier?

a) Public
b) Private
c) Protected
d) Void

**Answer : D**

Explanation: Public, private, protected and default are the access modifiers.

**Question 233:** Which of the following modifier means a particular variable cannot be accessed within the package?

a) private
b) public
c) protected
d) default

**Answer : A**

Explanation: Private variables are accessible only within the class.



**Question 234:** How can a protected modifier be accessed?

a) accessible only within the class
b) accessible only within package
c) accessible within package and outside the package but through inheritance only
d) accessible by all

**Answer : C**

Explanation: The protected access modifier is accessible within package and outside the package but only through inheritance. The protected access modifier can be used with data member, method and constructor. It cannot be applied in the class.

**Question 235:** What happens if constructor of class A is made private?

a) Any class can instantiate objects of class A
b) Objects of class A can be instantiated only within the class where it is declared
c) Inherited class can instantiate objects of class A
d) classes within the same package as class A can instantiate objects of class A

**Answer : B**

Explanation: If we make any class constructor private, we cannot create the instance of that class from outside the class.

**Question 236:** All the variables of interface should be?

a) default and final
b) default and static
c) public, static and final
d) protect, static and final

**Answer : C**

Explanation: Variables of an interface are public, static and final by default because the interfaces cannot be instantiated, final ensures the value assigned cannot be changed with the implementing class and public for it to be accessible by all the implementing classes.

**Question 237:** What is true of final class?

a) Final class cause compilation failure
b) Final class cannot be instantiated
c) Final class cause runtime failure
d) Final class cannot be inherited

**Answer : D**

Explanation: Final class cannot be inherited. This helps when we do not want classes to provide extension to these classes.

**Question 238:** How many copies of static and class variables are created when 10 objects are created of a class?

a) 1, 10
b) 10, 10
c) 10, 1
d) 1, 1

**Answer : A**

Explanation: Only one copy of static variables are created when a class is loaded. Each object instantiated has its own copy of instance variables.

**Question 239:** Which is the modifier when there is none mentioned explicitly?

a) protected
b) private
c) public
d) default

**Answer : D**

Explanation: Default is the access modifier when none is defined explicitly. It means the member (method or variable) can be accessed within the same package.

# Arrays Revisited & Keyword static

**Question 240:** Arrays in Java are implemented as?

a) class
b) object
c) variable
d) none of the mentioned

**Answer : B**

Explanation: None.

**Question 241:** Which of these keywords is used to prevent content of a variable from being modified?

a) final
b) last
c) constant
d) static

**Answer : A**

Explanation: A variable can be declared final, doing so prevents its content from being modified. Final variables must be initialized when it is declared.

**Question 242:** Which of these cannot be declared static?

a) class
b) object
c) variable
d) method

**Answer : B**

Explanation: static statements are run as soon as class containing then is loaded, prior to any object declaration.

**Question 243:** Which of the following statements are incorrect?

a) static methods can call other static methods only
b) static methods must only access static data
c) static methods can not refer to this or super in any way
d) when object of class is declared, each object contains its own copy of static variables

**Answer : D**

Explanation: All objects of class share same static variable, when object of a class are declared, all the objects share same copy of static members, no copy of static variables are made.

**Question 244:** Which of the following statements are incorrect?

a) Variables declared as final occupy memory
b) final variable must be initialized at the time of declaration
c) Arrays in java are implemented as an object
d) All arrays contain an attribute-length which contains the number of elements stored in the array

**Answer : A**

Explanation: None.

**Question 245:** Which of these methods must be made static?

a) main()
b) delete()
c) run()
d) finalize()

**Answer : A**

Explanation: main() method must be declared static, main() method is called by Java runtime system before any object of any class exists.

**Question 246:** What is the output of this program?

```
class access
{
   public int x;
   static int y;
   void cal(int a, int b)
   {
      x +=  a ;
      y +=  b;
   }
}
class static_specifier
{
   public static void main(String args[])
   {
      access obj1 = new access();
      access obj2 = new access();
      obj1.x = 0;
      obj1.y = 0;
      obj1.cal(1, 2);
      obj2.x = 0;
      obj2.cal(2, 3);
      System.out.println(obj1.x + " " + obj2.y);
   }
}
```

a) 1 2
b) 2 3
c) 3 2
d) 1 5

**Answer : D**

Explanation: None.

**Question 247:** What is the output of this program?

```
class access
{
  static int x;
  void increment()
  {
     x++;
  }
}
class static_use
{
   public static void main(String args[])
   {
      access obj1 = new access();
      access obj2 = new access();
      obj1.x = 0;
      obj1.increment();
      obj2.increment();
      System.out.println(obj1.x + " " + obj2.x);
   }
}
```

a) 1 2
b) 1 1
c) 2 2
d) Compilation Error

**Answer : C**

Explanation: All objects of class share same static variable, all the objects share same copy of static members, obj1.x and obj2.x refer to same element of class which has been incremented twice and its value is 2.

**Question 248:** What is the output of this program?

```
class static_out
{
   static int x;
   static int y;
   void add(int a , int b)
   {
      x = a + b;
      y = x + b;
   }
}
class static_use
{
   public static void main(String args[])
   {
      static_out obj1 = new static_out();
      static_out obj2 = new static_out();
      int a = 2;
      obj1.add(a, a + 1);
      obj2.add(5, a);
      System.out.println(obj1.x + " " + obj2.y);
   }
}
```

a) 7 7
b) 6 6
c) 7 9
d) 9 7

**Answer : C**

Explanation: None.



**Question 249:** What is the output of this program?

```
class Output
{
   public static void main(String args[])
   {
      int arr[] = {1, 2, 3, 4, 5};
      for ( int i = 0; i < arr.length - 2; ++i)
         System.out.println(arr[i] + " ");
   }
}
```

a) 1 2
b) 1 2 3
c) 1 2 3 4
d) 1 2 3 4 5

**Answer : B**

Explanation: arr.length() is 5, so the loop is executed for three times.

**Question 250:** What is the output of this program?

```
class Output
{
   public static void main(String args[])
   {
      int a1[] = new int[10];
      int a2[] = {1, 2, 3, 4, 5};
      System.out.println(a1.length + " " + a2.length);
   }
}
```

a) 10 5
b) 5 10
c) 0 10
d) 0 5

**Answer : A**

Explanation: Arrays in java are implemented as objects, they contain an attribute that is length which contains the number of elements that can be stored in the array. Hence a1.length gives 10 and a2.length gives 5.



**Question 251:** String in Java is a?

a) class
b) object
c) variable
d) character array

**Answer : A**

Explanation: None.

**Question 252:** Which method of String class is used to obtain character at specified index?

a) char()
b) Charat()
c) charat()
d) charAt()

**Answer : D**

Explanation: None.

**Question 253:** Which keyword is used to refer to member of base class from a subclass?

a) upper
b) super
c) this
d) none of the mentioned

**Answer : B**

Explanation: Whenever a subclass needs to refer to its immediate superclass, it can do so by use of the keyword super.

**Question 254:** Which method of String class can be used to test to strings for equality?

a) isequal()
b) isequals()
c) equal()
d) equals()

**Answer : D**

Explanation: None.

**Question 255:** Which of the following statements are incorrect?

a) String is a class
b) Strings in java are mutable
c) Every string is an object of class String
d) Java defines a peer class of String, called StringBuffer, which allows string to be altered

**Answer : B**

Explanation: Strings in Java are immutable that is they can not be modified.

**Question 256:** What is the output of this program?

```
class string_demo
{
   public static void main(String args[])
   {
      String obj = "I" + "like" + "Java";
      System.out.println(obj);
   }
}
```

a) I
b) like
c) Java
d) IlikeJava

**Answer : D**

Explanation: Java defines an operator +, it is used to concatenate strings.

**Question 257:** What is the output of this program?

```
class string_class
{
   public static void main(String args[])
   {
      String obj = "I LIKE JAVA";
      System.out.println(obj.charAt(3));
   }
}
```

a) I
b) L
c) K
d) E

**Answer : A**

Explanation: charAt() is a method of class String which gives the character specified by the index. obj.charAt(3) gives 4th character i:e I.

**Question 258:** What is the output of this program?

```
class string_class
{
   public static void main(String args[])
   {
      String obj = "I LIKE JAVA";
      System.out.println(obj.length());
   }
}
```

a) 9
b) 10
c) 11
d) 12

**Answer : C**

Explanation: None.

**Question 259:** What is the output of this program?

```
class string_class
{
   public static void main(String args[])
   {
      String obj = "hello";
      String obj1 = "world";
      String obj2 = obj;
      obj2 = " world";
      System.out.println(obj + " " + obj2);
   }
}
```

a) hello hello
b) world world
c) hello world
d) world hello

**Answer : C**

Explanation: None.

**Question 260:** What is the output of this program?

```
class string_class
{
   public static void main(String args[])
   {
      String obj = "hello";
      String obj1 = "world";
      String obj2 = "hello";
      System.out.println(obj.equals(obj1) + " " + obj.equals(obj2));
   }
}
```

a) false false
b) true true
c) true false
d) false true

**Answer : D**

Explanation: equals() is method of class String, it is used to check equality of two String objects, if they are equal, true is retuned else false.

# Methods Taking Parameters

**Question 261:** What is the process of defining more than one method in a class differentiated by parameters?

a) Function overriding
b) Function overloading
c) Function doubling
d) None of the mentioned

**Answer : B**

Explanation: Function overloading is a process of defining more than one method in a class with same name differentiated by function signature i:e return type or parameters type and number. Example – int volume(int length, int width) & int volume(int length, int width , int height) can be used to calculate volume.

**Question 262:** Which of the following can be used to differentiate two or more methods having the same name?

a) Parameters data type
b) Number of parameters
c) Return type of method
d) All of the mentioned

**Answer : D**

Explanation: None.

**Question 263:** Which statement is incorrect?

a) Two or more methods with same name can be differentiated on the basis of their parameters data type
b) Two or more method having same name can be differentiated on basis of number of parameters
c) Any already defined method in java library can be defined again in the program with different data type of parameters
d) If a method is returning a value the calling statement must have a variable to store that value

**Answer : D**

Explanation: Even if a method is returning a value, it is not necessary to store that value.

**Question 264:** What is the output of this program?

```
class box
{
   int width;
   int height;
   int length;
   int volume;
   void volume(int height, int length, int width)
   {
       volume = width * height * length;
   }
}
class Prameterized_method{
   public static void main(String args[])
   {
      box obj = new box();
      obj.height = 1;
      obj.length = 5;
      obj.width = 5;
      obj.volume(3, 2, 1);
      System.out.println(obj.volume);
   }
}
```

a) 0
b) 1
c) 6
d) 25

**Answer : C**

Explanation: None.



**Question 265:** What is the output of this program?

```
class equality
{
   int x;
   int y;
   boolean isequal()
   {
      return(x == y);
   }
}
```

```
class Output
{
   public static void main(String args[])
   {
      equality obj = new equality();
      obj.x = 5;
      obj.y = 5;
      System.out.println(obj.isequal);
   }
}
```

a) false
b) true
c) 0
d) 1

**Answer : B**

Explanation: None.

**Question 266:** What is the output of this program?

```
class box
{
   int width;
   int height;
   int length;
   int volume;
   void volume()
   {
      volume = width * height * length;
   }
   void volume(int x)
   {
      volume = x;
   }
}
class Output
{
   public static void main(String args[])
   {
      box obj = new box();
      obj.height = 1;
      obj.length = 5;
      obj.width = 5;
      obj.volume(5);
      System.out.println(obj.volume);
   }
}
```

a) 0
b) 5
c) 25
d) 26

**Answer : B**

Explanation: None.

**Question 267:** What is the output of this program?

```
class Output
{
   static void main(String args[])
   {
       int x , y = 1;
       x = 10;
       if(x != 10 && x / 0 == 0)
          System.out.println(y);
       else
          System.out.println(++y);
   }
}
```

a) 1
b) 2
c) Runtime Error
d) Compilation Error

**Answer : D**

Explanation: main() method must be made public. Without main() being public java run time system will not be able to access main() and will not be able to execute the code.

**Question 268:** What is the output of this program?

```
class area
{
   int width;
   int length;
   int height;
   area()
   {
   width = 5;
   length = 6;
   height = 1;
   }
   void volume()
   {
       volume = width * height * length;
   }
}
```

```
class cons_method
{
   public static void main(String args[])
   {
      area obj = new area();
      obj.volume();
      System.out.println(obj.volume);
   }
}
```

a) 0
b) 1
c) 25
d) 30

**Answer : D**

Explanation: None.

# Command Line Arguments

**Question 269:** Which of this method is given parameter via command line arguments?

a) main()
b) recursive() method
c) Any method
d) System defined methods

**Answer : A**

Explanation: Only main() method can be given parameters via using command line arguments.

**Question 270:** Which data types is used to store command line arguments?

a) Array
b) Stack
c) String
d) Integer

**Answer : C**

Explanation: None.

**Question 271:** How many arguments can be passed to main()?

a) Infinite
b) Only 1
c) System Dependent
d) None of the mentioned

**Answer : A**

Explanation: None.

**Question 272:** Which is a correct statement about args in this line of code?
public static void main(String args[])

a) args is a String
b) args is a Character
c) args is an array of String
d) args in an array of Character

**Answer : C**

Explanation: args in an array of String.

**Question 273:** What is the output of this program, Command line execution is done as – "java Output This is a command Line"?

```
class Output
{
   public static void main(String args[])
   {
      System.out.print("args[0]");
   }
}
```

a) java
b) Output
c) This
d) is

**Answer : C**

Explanation: None.

**Question 274:** What is the output of this program, Command line exceution is done as – "java Output This is a command Line"?

```
class Output
{
   public static void main(String args[])
   {
      System.out.print("args[3]");
   }
}
```

a) java
b) is
c) This
d) command

**Answer : D**

Explanation: None.



**Question 275:** What is the output of this program, Command line execution is done as – "java Output This is a command Line"?

```
class Output
{
   public static void main(String args[])
   {
      System.out.print("args");
   }
}
```

a) This
b) java Output This is a command Line
c) This is a command Line
d) Compilation Error

**Answer : C**

Explanation: None.

**Question 276:** Which of these access specifiers must be used for main() method?

a) private
b) public
c) protected
d) none of the mentioned

**Answer : B**

Explanation: main() method must be specified public as it called by Java run time system, outside of the program. If no access specifier is used then by default member is public within its own package & cannot be accessed by Java run time system.

**Question 277:** What would be the output of the following snippet, if attempted to compile and run this code with command line argument "java abc Rakesh Sharma"?

```
public class abc
{
    int a=2000;
    public static void main(String argv[])
    {
       System.out.println(argv[1]+" :-Please pay Rs."+a);
    }
}
```

a) Compile time error
b) Compilation but runtime error
c) Rakesh :-Please pay Rs.2000
d) Sharma :-Please pay Rs.2000

**Answer : A**

Explanation: Main method is static and cannot access non static variable a

**Question 278:** What would be the output of following snippet, if attempted to compile and execute?

```
class abc
{
  public static void main(String args[])
  {
    if(args.length>0)
        System.out.println(args.length);
  }
}
```

a) The snippet compiles, runs and prints 0
b) The snippet compiles, runs and prints 1
c) The snippet does not compile
d) The snippet compiles and runs but does not print anything

**Answer : D**

Explanation: As no argument is passed to the code, the length of args is 0. So the code will not print.

**Question 279:** What would be the output of following snippet, if compiled and executed with command line argument "java abc 1 2 3"?

```
public class abc
{
  static public void main(String [] xyz)
  {
    for(int n=1;n<xyz.length; n++)
    {
      System.out.println(xyz[n]+"");
    }
  }
}
```

a) 1 2
b) 2 3
c) 1 2 3
d) Compilation error

**Answer : B**

Explanation: The index of array starts with 0. Since the loop is starting with 1 it will print 2 3.

**Question 280:** What is the output of the following snippet running with "java demo I write java code"?

```
public class demo
{
  public static void main(String args[])
  {
      System.out.println(args[0]+""+args[args.length-1]);
  }
}
```

a) The snippet compiles, runs and prints "java demo"
b) The snippet compiles, runs and prints "java code"
c) The snippet compiles, runs and prints "demo code"
d) The snippet compiles, runs and prints "I code"

**Answer : D**

Explanation: The index of array starts with 0 till length – 1. Hence it would print "I code".

**Question 281:** What would be the output of the following snippet, if compiled and executed with command line "hello there"?

```
public class abc
{
      String[] xyz;
      public static void main(String argv[])
      {
          xyz=argv;
     }
     public void runMethod()
     {
          System.out.println(argv[1]);
     }
}
```

a) Compile time error
b) Output would be "hello"
c) Output would be "there"
d) Output would be "hello there"

**Answer : A**

Explanation: Error would be "Cannot make static reference to a non static variable". Even if main method was not static, the array argv is local to the main method and would not be visible within runMethod.

**Question 282:** Which annotation is used to represent command line input and assigned to correct data type?

a) @Input
b) @Variable
c) @Command Line
d) @Parameter

**Answer : D**

Explanation: @Parameter, @Parameter(names = { "-log", "-verbose" }, description = "Level of verbosity"), etc are various forms of using @Parameter

# Recursion

**Question 283:** Which of the following is used by operating system to manage the Recursion in Java?

a) Array
b) Stack
c) Queue
d) Tree

**Answer : B**

Explanation: Recursions are always managed by using stack.

**Question 284:** Which of these is not a correct statement?

a) A recursive method must have a base case
b) Recursion always uses stack
c) Recursive methods are faster that programmers written loop to call the function repeatedly using a stack
d) Recursion is managed by Java Runtime environment

**Answer : D**

Explanation: Recursion is always managed by operating system.

**Question 285:** What is the output of this program?

```
class recursion
{
   int func (int n)
   {
      int result;
      result = func (n - 1);
      return result;
   }
}
class Output
{
   public static void main(String args[])
   {
      recursion obj = new recursion() ;
      System.out.print(obj.func(12));
   }
}
```

a) 0
b) 1
c) Compilation Error
d) Runtime Error

**Answer : D**

Explanation: Since the base case of the recursive function func() is not defined hence infinite loop occurs and results in Stack Overflow.

**Question 286:** What is the output of this program?

```
class recursion
{
   int func (int n)
   {
      int result;
      if (n == 1)
         return 1;
      result = func (n - 1);
      return result;
   }
}
class Output
{
   public static void main(String args[])
   {
      recursion obj = new recursion() ;
      System.out.print(obj.func(5));
   }
}
```

a) 0
b) 1
c) 120
d) None of the mentioned

**Answer : B**

Explanation: None.



**Question 287:** What is the output of this program?

```
class recursion
{
   int fact(int n)
   {
      int result;
      if (n == 1)
         return 1;
      result = fact(n - 1) * n;
      return result;
   }
}
```

```
class Output
{
   public static void main(String args[])
   {
      recursion obj = new recursion() ;
      System.out.print(obj.fact(5));
   }
}
```

a) 24
b) 30
c) 120
d) 720

**Answer : C**

Explanation: fact() method recursively calculates factorial of a number, when value of n reaches 1, base case is excuted and 1 is returned.

**Question 288:** What is the output of this program?

```
class recursion
{
   int fact(int n)
   {
      int result;
      if (n == 1)
         return 1;
      result = fact(n - 1) * n;
      return result;
   }
}

class Output
{
   public static void main(String args[])
   {
      recursion obj = new recursion() ;
      System.out.print(obj.fact(1));
   }
}
```

a) 1
b) 30
c) 120
d) Runtime Error

**Answer : A**

Explanation: fact() method recursively calculates factorial of a number, when value of n reaches 1, base case is excuted and 1 is returned.

**Question 289:** What is the output of this program?

```
class recursion
{
   int fact(int n)
   {
      int result;
      if (n == 1)
         return 1;
      result = fact(n - 1) * n;
      return result;
   }
}
class Output
{
   public static void main(String args[])
   {
      recursion obj = new recursion() ;
      System.out.print(obj.fact(6));
   }
}
```

a) 1
b) 30
c) 120
d) 720

**Answer : D**

Explanation: None.

# Method overriding

**Question 290:** Which of this keyword can be used in a subclass to call the constructor of superclass?

a) super
b) this
c) extent
d) extends

**Answer : A**

Explanation: None.

**Question 291:** What is the process of defining a method in a subclass having same name & type signature as a method in its superclass?

a) Method overloading
b) Method overriding
c) Method hiding
d) None of the mentioned

**Answer : B**

Explanation: None.

**Question 292:** Which keyword can be used to prevent Method overriding?

a) static
b) constant
c) protected
d) final

**Answer : D**

Explanation: To disallow a method from being overridden, specify final as a modifier at the start of its declaration. Methods declared as final cannot be overridden.

**Question 293:** Which is the correct way of calling a constructor having no parameters, of superclass A by subclass B?

a) super(void);
b) superclass.();
c) super.A();
d) super();

**Answer : D**

Explanation: None.

**Question 294:** At line number 2 below, choose 3 valid data-type attributes/qualifiers among "final, static, native, public, private, abstract, protected"

```
public interface Status
  {
     /* insert qualifier here */ int MY_VALUE = 10;
  }
```

a) final, native, private
b) final, static, protected
c) final, private, abstract
d) final, static, public

**Answer : D**

Explanation: Every interface variable is implicitly public static and final.

**Question 295:** Which concept supports method overriding in Java?

a) Abstraction
b) Encapsulation
c) Polymorphism
d) None of the mentioned

**Answer : C**

Explanation: None.

**Question 296:** What is the output of this program?

```
class Alligator
{
    public static void main(String[] args)
    {
        int [][]x[] = {{1,2}, {3,4,5}, {6,7,8,9}};
        int [][]y = x;
            System.out.println(y[2][1]);
    }
}
```

a) 2
b) 3
c) 7
d) Compilation Error

**Answer : C**

Explanation: Both x,and y are pointing to the same array.

**Question 297:** What is the output of this program?

```
final class A
{
    int i;
}
class B extends A
{
   int j;
   System.out.println(j + " " + i);
}
class inheritance
{
   public static void main(String args[])
   {
      B obj = new B();
      obj.display();
   }
}
```

a) 2 2
b) 3 3
c) Runtime Error
d) Compilation Error

**Answer : D**

Explanation: class A has been declared final hence it cannot be inherited by any other class. Hence class B does not have member i, giving compilation error.

**Question 298:** What is the output of this program?

```
class Abc
{
   public static void main(String[]args)
   {
      String[] elements = { "for", "tea", "too" };
      String first = (elements.length > 0) ? elements[0]: null;
   }
}
```

a) Compilation error
b) An exception is thrown at run time
c) The variable first is set to null
d) The variable first is set to elements[0].

**Answer : D**

Explanation: The value at the 0th position will be assigned to the variable first.

**Question 299:** What is the output of this program?

```
class A
{
   int i;
   public void display()
   {
      System.out.println(i);
   }
}
class B extends A
{
   int j;
   public void display()
   {
      System.out.println(j);
   }
}
class Dynamic_dispatch
{
   public static void main(String args[])
   {
      B obj2 = new B();
      obj2.i = 1;
      obj2.j = 2;
      A r;
      r = obj2;
      r.display();
   }
}
```

a) 1
b) 2
c) 3
d) 4

**Answer : B**

Explanation: r is reference of type A, the program assigns a reference of object obj2 to r and uses that reference to call function display() of class B.

# The Object Class

**Question 300:** Which class is superclass of every class in Java?

a) String class
b) Object class
c) Abstract class
d) ArrayList class

**Answer : B**

Explanation: Object class is superclass of every class in Java.

**Question 301:** Which method of Object class can clone an object?

a) Objectcopy()
b) copy()
c) Object clone()
d) clone()

**Answer : C**

Explanation: None.

**Question 302:** Which method of Object class is used to obtain class of an object at run time?

a) get()
b) void getclass()
c) Class getclass()
d) None of the mentioned

**Answer : C**

Explanation: None.



**Question 303:** Which keyword cannot be used for a class which has been declared final?

a) abstract
b) extends
c) abstract and extends
d) none of the mentioned

**Answer : A**

Explanation: A abstract class is incomplete by itself and relies upon its subclasses to provide complete implementation. If we declare a class final then no class can inherit that class, an abstract class needs its subclasses hence both final and abstract cannot be used for a same class.

**Question 304:** Which class relies upon its subclasses for complete implementation of its methods?

a) Object class
b) abstract class
c) ArrayList class
d) None of the mentioned

**Answer : B**

Explanation: None.

**Question 305:** What is the output of this program?

```
abstract class A
{
   int i;
   abstract void display();
}
class B extends A
{
   int j;
   void display()
   {
      System.out.println(j);
   }
}
class Abstract_demo
{
   public static void main(String args[])
   {
      B obj = new B();
      obj.j=2;
      obj.display();
   }
}
```

a) 0
b) 2
c) Runtime Error
d) Compilation Error

**Answer : B**

Explanation: class A is an abstract class, it contains a abstract function display(), the full implementation of display() method is given in its subclass B, Both the display functions are the same. Prototype of display() is defined in class A and its implementation is given in class B.

**Question 306:** What is the output of this program?

```
class A
{
     int i;
     int j;
    A()
    {
       i = 1;
       j = 2;
    }
}
class Output
{
    public static void main(String args[])
    {
        A obj1 = new A();
        A obj2 = new A();
         System.out.print(obj1.equals(obj2));
    }
}
```

a) false
b) true
c) 1
d) Compilation Error

**Answer : A**

Explanation: obj1 and obj2 are two different objects. equals() is a method of Object class, Since Object class is superclass of every class it is available to every object.

**Question 307:** What is the output of this program?

```
class Output
{
   public static void main(String args[])
   {
        Object obj = new Object();
         System.out.print(obj.getclass());
   }
}
```

a) Object
b) class Object
c) class java.lang.Object
d) Compilation Error

**Answer : C**

Explanation: None.

**Question 308:** What is the output of this program?

```
class A
{
    int i;
    int j;
    A()
    {
        i = 1;
        j = 2;
    }
}
class Output
{
    public static void main(String args[])
    {
        A obj1 = new A();
        System.out.print(obj1.toString());
    }
}
```

a) true
b) false
c) String associated with obj1
d) Compilation Error

**Answer : C**

Explanation: toString() is method of class Object, since it is superclass of every class, every object has this method. toString() returns the string associated with the calling object.

# Inheritance – Abstract Class and Super

**Question 309:** Which keyword is used to define an abstract class?

a) abst
b) abstract
c) Abstract
d) abstract class

**Answer : B**

Explanation: None.

**Question 310:** If a class inheriting an abstract class does not define all of its function then it will be known as?

a) Abstract
b) A simple class
c) Static class
d) None of the mentioned

**Answer : A**

Explanation: Any subclass of an abstract class must either implement all of the abstract method in the superclass or be itself declared abstract.

**Question 311:** Which is not a correct statement?

a) Every class containing abstract method must be declared abstract
b) Abstract class defines only the structure of the class not its implementation
c) Abstract class can be initiated by new operator
d) Abstract class can be inherited

**Answer : C**

Explanation: Abstract class cannot be directly initiated with new operator, Since abstract class does not contain any definition of implementation it is not possible to create an abstract object.

**Question 312:** Which package contains abstract keyword?

a) java.lang
b) java.util
c) java.io
d) java.system

**Answer : A**

Explanation: None.

**Question 313:** What is the output of this program?

```
class A
{
   public int i;
   private int j;
}
class B extends A
{
   void display()
   {
      super.j = super.i + 1;
      System.out.println(super.i + " " + super.j);
   }
}
class inheritance
{
   public static void main(String args[])
   {
      B obj = new B();
      obj.i=1;
      obj.j=2;
      obj.display();
   }
}
```

a) 2 2
b) 3 3
c) Runtime Error
d) Compilation Error

**Answer : D**

Explanation: Class contains a private member variable j, this cannot be inherited by subclass B and does not have access to it.



**Question 314:** What is the output of this program?

```
class A
{
   public int i;
   public int j;
   A()
   {
      i = 1;
      j = 2;
   }
}
```

```
class B extends A
{
   int a;
      B()
   {
      super();
   }
}
class super_use
{
   public static void main(String args[])
   {
      B obj = new B();
      System.out.println(obj.i + " " + obj.j)
   }
}
```

a) 1 2
b) 2 1
c) Runtime Error
d) Compilation Error

**Answer : A**

Explanation: Keyword super is used to call constructor of class A by constructor of class B. Constructor of a initializes i & j to 1 & 2 respectively.



**Question 315:** What is the output of this program?

```
class A
{
   int i;
   void display()
   {
      System.out.println(i);
   }
}
class B extends A
{
   int j;
   void display()
   {
      System.out.println(j);
   }
}
```

```
class method_overriding
{
   public static void main(String args[])
   {
      B obj = new B();
      obj.i=1;
      obj.j=2;
      obj.display();
   }
}
```

a) 0
b) 1
c) 2
d) Compilation Error

**Answer : C**

Explanation: class A & class B both contain display() method, class B inherits class A, when display() method is called by object of class B, display() method of class B is executed rather than that of Class A.

**Question 316:** What is the output of this program?

```
class A
{
   public int i;
   protected int j;
}
class B extends A
{
   int j;
   void display()
   {
      super.j = 3;
      System.out.println(i + " " + j);
   }
}
class Output
{
   public static void main(String args[])
   {
      B obj = new B();
      obj.i=1;
      obj.j=2;
      obj.display();
   }
}
```

a) 1 2
b) 2 1
c) 1 3
d) 3 1

**Answer : A**

Explanation: Both class A & B have member with same name that is j, member of class B will be called by default if no specifier is used. I contains 1 & j contains 2, printing 1 2.

# Inheritance

**Question 317:** Which of this keyword must be used to inherit a class?

a) super
b) this
c) extent
d) extends

**Answer : D**

Explanation: None.

**Question 318:** A class member declared protected becomes a member of subclass of which type?

a) public member
b) private member
c) protected member
d) static member

**Answer : B**

Explanation: A class member declared protected becomes a private member of subclass.

**Question 319:** Which of this is correct way of inheriting class A by class B?

a) class B + class A {}
b) class B inherits class A {}
c) class B extends A {}
d) class B extends class A {}

**Answer : C**

Explanation: None.

**Question 320:** Which two classes use the Shape class correctly?

A. public class Circle implements Shape { private int radius; }
B. public abstract class Circle extends Shape { private int radius; }
C. public class Circle extends Shape
```
{
  private int radius;
  public void draw();
}
```
D. public abstract class Circle implements Shape
```
{
  private int radius;
  public void draw();
}
```

E.
```
public class Circle extends Shape
  {
    private int radius;
    public void draw()
    {
      /* code here */
    }
  }
```
F.
```
public abstract class Circle implements Shape
  {
    private int radius;
    public void draw()
    {
      /* code here */
    }
  }
```

a) B,E
b) A,C
c) C,E
d) T,H

**Answer : A**

Explanation: If one is extending any class, then they should use extends keyword not implements.

**Question 321:** What is the output of this program?

```
class A
{
   int i;
   void display()
   {
      System.out.println(i);
   }
}
class B extends A
{
   int j;
   void display()
   {
      System.out.println(j);
   }
}
```

```
class inheritance_demo
{
   public static void main(String args[])
   {
      B obj = new B();
      obj.i=1;
      obj.j=2;
      obj.display();
   }
}
```

a) 0
b) 1
c) 2
d) Compilation Error

**Answer : C**

Explanation: Class A & class B both contain display() method, class B inherits class A, when display() method is called by object of class B, display() method of class B is executed rather than that of Class A.

**Question 322:** What is the output of this program?

```
class A
{
   int i;
}
class B extends A
{
   int j;
   void display()
   {
      super.i = j + 1;
      System.out.println(j + " " + i);
   }
}
class inheritance
{
   public static void main(String args[])
   {
      B obj = new B();
      obj.i=1;
      obj.j=2;
      obj.display();
   }
}
```

a) 2 2
b) 3 3
c) 2 3
d) 3 2

**Answer : C**

Explanation: None.

**Question 323:** What is the output of this program?

```
class A
{
   public int i;
   public int j;
   A()
   {
      i = 1;
      j = 2;
   }
}
class B extends A
{
   int a;
   B()
   {
      super();
   }
}
class super_use
{
   public static void main(String args[])
   {
      B obj = new B();
      System.out.println(obj.i + " " + obj.j)
   }
}
```

a) 1 2
b) 2 1
c) Runtime Error
d) Compilation Error

**Answer : A**

Explanation: Keyword super is used to call constructor of class A by constructor of class B. Constructor of a initializes i & j to 1 & 2 respectively.

**Question 324:** What is not type of inheritance?

a) Single inheritance
b) Double inheritance
c) Hierarchical inheritance
d) Multiple inheritance

**Answer : B**

Explanation: Inheritance is way of acquiring attributes and methods of parent class. Java supports hierarchical inheritance directly.

**Question 325:** Using which of the following, multiple inheritance in Java can be implemented?

a) Interfaces
b) Multithreading
c) Protected methods
d) Private methods

**Answer : A**

Explanation: Multiple inheritance in java is implemented using interfaces. Multiple interfaces can be implemented by a class.

**Question 326:** All classes in Java are inherited from which class?

a) java.lang.class
b) java.class.inherited
c) java.class.object
d) java.lang.Object

**Answer : D**

Explanation: All classes in Java are inherited from Object class. Interfaces are not inherited from Object Class.

**Question 327:** In order to restrict a variable of a class from inheriting to subclass, how variable should be declared?

a) Protected
b) Private
c) Public
d) Static

**Answer : B**

Explanation: By declaring variable private, the variable will not be available in inherited to subclass.

**Question 328:** If super class and subclass have same variable name, which keyword should be used to use super class?

a) super
b) this
c) upper
d) classname

**Answer : A**

Explanation: Super keyword is used to access hidden super class variable in subclass.

**Question 329:** Static members are not inherited to subclass.

a) True
b) False

**Answer : B**

Explanation: Static members are also inherited to subclasses.

**Question 330:** Which of the following is used for implementing inheritance through an interface?

a) inherited
b) using
c) extends
d) implements

**Answer : D**

Explanation: Interface is implemented using implements keyword. A concrete class must implement all the methods of an interface, else it must be declared abstract.

**Question 331:** Which of the following is used for implementing inheritance through class?

a) inherited
b) using
c) extends
d) implements

**Answer : C**

Explanation: Class can be extended using extends keyword. One class can extend only one class. A final class cannot be extended.

**Question 332:** What would be the result if a class extends two interfaces and both have a method with same name and signature?

a) Runtime error
b) Compile time error
c) Code runs successfully
d) First called method is executed successfully

**Answer : B**

Explanation: In case of such conflict, compiler will not be able to link a method call due to ambiguity. It will throw compile time error.

# String Handling Basics

**Question 333:** Which class is superclass of String and StringBuffer class?

a) java.util
b) java.lang
c) ArrayList
d) None of the mentioned

**Answer : B**

Explanation: None.

**Question 334:** 2. Which operator can be used to concatenate two or more String objects?

a) +
b) +=
c) &
d) ||

**Answer : A**

Explanation: Operator + is used to concatenate strings, Example String s = "i " + "like " + "java"; String s contains "I like java".

**Question 335:** Which of this method of class String is used to obtain a length of String object?

a) get()
b) Sizeof()
c) lengthof()
d) length()

**Answer : D**

Explanation: Method length() of string class is used to get the length of the object which invoked method length().

**Question 336:** Which constructor is used to create an empty String object?

a) String()
b) String(void)
c) String(0)
d) None of the mentioned

**Answer : A**

Explanation: None.

**Question 337:** What is the output of this program?

```
class String_demo
{
   public static void main(String args[])
   {
      char chars[] = {'a', 'b', 'c'};
      String s = new String(chars);
      System.out.println(s);
   }
}
```

a) a
b) b
c) c
d) abc

**Answer : D**

Explanation: String(chars) is a constructor of class string, it initializes string s with the values stored in character array chars, therefore s contains "abc".

**Question 338:** What is the output of this program?

```
class String_demo
{
   public static void main(String args[])
   {
      int ascii[] = { 65, 66, 67, 68};
      String s = new String(ascii, 1, 3);
      System.out.println(s);
   }
}
```

a) ABC
b) BCD
c) CDA
d) ABCD

**Answer : B**

Explanation: ascii is an array of integers which contains ascii codes of Characters A, B, C, D. String(ascii, 1, 3) is an constructor which initializes s with Characters corresponding to ascii codes stored in array ascii, starting position being given by 1 & ending position by 3, Thus s stores BCD.

**Question 339:** What is the output of this program?

```
class String_demo
{
   public static void main(String args[])
   {
      char chars[] = {'a', 'b', 'c'};
      String s = new String(chars);
      String s1 = "abcd";
      int len1 = s1.length();
      int len2 = s.length();
      System.out.println(len1 + " " + len2);
   }
}
```

a) 3 0
b) 0 3
c) 3 4
d) 4 3

**Answer : D**

Explanation: None.

**Question 340:** Which constructor is used to create an empty String object?

a) String()
b) String(void)
c) String(0)
d) None of the mentioned

**Answer : A**

Explanation: None.

## Character Extraction

**Question 341:** Which method of class String is used to extract more than one character at a time a String object?

a) getchars()
b) GetChars()
c) Getchars()
d) getChars()

**Answer : D**

Explanation: None.

**Question 342:** What will be output of the following code?

```
public class Boxer1
{
   Integer i;
   int x;
  public Boxer1(int y)
  {
      x = i+y;
      System.out.println(x);
  }
  public static void main(String[]args)
  {
     new Boxer1 (new Integer(4));
  }
}
```

a) The value “4” is printed at the command line
b) Compilation fails because of an error in line
c) A NullPointerException occurs at runtime
d) An IllegalStateException occurs at runtime

**Answer : D**

Explanation: Because we are performing operation on reference variable which is null.

**Question 343:** Which method can be used to convert all characters in a String into a character array?

a) charAt()
b) both getChars() & charAt()
c) both toCharArray() & getChars()
d) all of the mentioned

**Answer : C**

Explanation: charAt() return one character only not array of character.

**Question 344:** What is the output of this program?

```
class output
{
   public static void main(String args[])
   {
      String c = "Hello i love java";
      int start = 2;
      int end = 9;
      char s[]=new char[end-start];
      c.getChars(start,end,s,0);
      System.out.println(s);
   }
}
```

a) Hello, i love java
b) i love ja
c) lo i lo
d) llo i l

**Answer : D**

Explanation: getChars(start,end,s,0) returns an array from the string c, starting index of array is pointed by start and ending index is pointed by end, s is the target character array where the new string of letters is going to be stored and the new string will be stored from 0th position in s.

**Question 345:** Which method is an alternative to getChars() that stores the characters in an array of bytes?

a) getBytes()
b) GetByte()
c) giveByte()
d) GiveByte()

**Answer : A**

Explanation: getBytes() stores the character in an array of bytes. It uses default character to byte conversions provided by the platform.

**Question 346:** What is the output of this program?

```
class output
{
   public static void main(String args[])
   {
      String a = "hello i love java";
      System.out.println(a.indexOf('i')+" "+a.indexOf('o') +"  "+a.lastIndexOf('i')+" "+a.lastIndexOf('o'));
   }
}
```

a) 6 4 6 9
b) 5 4 5 9
c) 7 8 8 9
d) 4 3 6 9

**Answer : A**

Explanation: indexof('c') and lastIndexof('c') are pre defined function which are used to get the index of first and last occurrence of the character pointed by c in the given array.

**Question 347:** What is the output of this program?

```
class output
{
   public static void main(String args[])
   {
      char c[]={'a', '1', 'b' , ' ' ,'A' , '0'};
      for (int i = 0; i < 5; ++i)
      {
           if(Character.isDigit(c[i]))
              System.out.println(c[i]+" is a digit");
           if(Character.isWhitespace(c[i]))
              System.out.println(c[i]+" is a Whitespace character");
           if(Character.isUpperCase(c[i]))
              System.out.println(c[i]+" is an Upper case Letter");
           if(Character.isLowerCase(c[i]))
              System.out.println(c[i]+" is a lower case Letter");
        i=i+3;
      }
   }
}
```

a)
a is a lower case Letter
is White space character

b)
b is a lower case Letter
is White space character

c)
a is a lower case Letter
A is an upper case Letter

d)
a is a lower case Letter
0 is a digit

**Answer : C**

Explanation: Character.isDigit(c[i]),Character.isUpperCase(c[i]),Character.isWhitespace(c[i]) are the function of library java.lang. They are used to find weather the given character is of specified type or not. They return true or false i:e Boolean variable.

**Question 348:** What is the output of this program?

```
class output
{
   public static void main(String args[])
   {
      char ch;
      ch = "hello".charAt(1);
      System.out.println(ch);
   }
}
```

a) h
b) e
c) l
d) o

**Answer : B**

Explanation: “hello” is a String literal, method charAt() returns the character specified at the index position. Character at index position 1 is e of hello, hence ch contains e.

# String Comparison

**Question 349:** Which method of class String is used to compare two String objects for their equality?

a) equals()
b) Equals()
c) isequal()
d) Isequal()

**Answer : A**

Explanation: None.

**Question 350:** Which method is used to compare a specific region inside a string with another specific region in another string?

a) regionMatch()
b) match()
c) RegionMatches()
d) regionMatches()

**Answer : D**

Explanation: None.

**Question 351:** Which method of class String is used to check whether a given object starts with a particular string literal?

a) startsWith()
b) endsWith()
c) Starts()
d) ends()

**Answer : A**

Explanation: Method startsWith() of string class is used to check whether the String in question starts with a specified string. It is a specialized form of method regionMatches().

**Question 352:** What is the value returned by function compareTo() if the invoking string is less than the string compared?

a) zero
b) value less than zero
c) value greater than zero
d) none of the mentioned

**Answer : B**

Explanation: compareTo() function returns zero when both the strings are equal, it returns a value less than zero if the invoking string is less than the other string being compared and value greater than zero when invoking string is greater than the string compared to.

**Question 353:** Which data type value is returned by equals() method of String class?

a) char
b) int
c) boolean
d) all of the mentioned

**Answer : C**

Explanation: equals() method of string class returns boolean value true if both the string are equal and false if they are unequal.

**Question 354:** What is the output of this program?

```
class output
{
   public static void main(String args[])
   {
      String c = "Hello i love java";
      boolean var;
      var = c.startsWith("hello");
      System.out.println(var);
   }
}
```

a) true
b) false
c) 0
d) 1

**Answer : B**

Explanation: startsWith() method is case sensitive “hello” and “Hello” are treated differently, hence false is stored in var.



**Question 355:** What is the output of this program?

```
class output
{
   public static void main(String args[])
   {
      String s1 = "Hello i love java";
      String s2 = new String(s1);
      System.out.println((s1 == s2) + " " + s1.equals(s2));
   }
}
```

a) true true
b) false false
c) true false
d) false true

**Answer : D**

Explanation: The == operator compares two object references to see whether they refer to the same instance, where as equals() compares the content of the two objects.

**Question 356:** What is the output of this program?

```
class output
{
   public static void main(String args[])
   {
      String s1 = "Hello";
      String s2 = new String(s1);
      String s3 = "HELLO";
      System.out.println(s1.equals(s2) + " " + s2.equals(s3));
   }
}
```

a) true true
b) false false
c) true false
d) false true

**Answer : C**

Explanation: None.



**Question 357:** In the below code which code fragment should be inserted at line 3 so that the output will be: "123abc 123abc"?

```
1 StringBuilder sb1 = new StringBuilder("123");
2 String s1 = "123";
3  // insert code here
4 System.out.println(sb1 + " " + s1);
```

a) sb1.append("abc"); s1.append("abc");
b) sb1.append("abc"); s1.concat("abc");
c) sb1.concat("abc"); s1.append("abc");
d) sb1.append("abc"); s1 = s1.concat("abc");

**Answer : D**

Explanation: append() is stringbuffer method and concat is String class method. append() is stringbuffer method and concat is String class method.

**Question 358:** What is the output of this program?

```
class output
{
   public static void main(String args[])
   {
       String chars[] = {"a", "b", "c", "a", "c"};
       for (int i = 0; i < chars.length; ++i)
          for (int j = i + 1; j < chars.length; ++j)
             if(chars[i].compareTo(chars[j]) == 0)
                System.out.print(chars[j]);
   }
}
```

a) ab
b) bc
c) ca
d) ac

**Answer : D**

Explanation: compareTo() function returns zero when both the strings are equal, it returns a value less than zero if the invoking string is less than the other string being compared and value greater than zero when invoking string is greater than the string compared to.

# Searching & Modifying a String

**Question 359:** Which of this method of class String is used to extract a substring from a String object?

a) substring()
b) Substring()
c) SubString()
d) None of the mentioned

**Answer : A**

Explanation: None.

**Question 360:** What will s2 contain after following lines of code?

```
String s1 = "one";
String s2 = s1.concat("two")
```

a) one
b) two
c) onetwo
d) twoone

**Answer : C**

Explanation: Two strings can be concatenated by using concat() method.

**Question 361:** Which method of class String is used to remove leading and trailing whitespaces?

a) startsWith()
b) trim()
c) Trim()
d) doTrim()

**Answer : B**

Explanation: None.



**Question 362:** Which of the following statement is correct?

a) replace() method replaces all occurrences of one character in invoking string with another character
b) replace() method replaces only first occurrences of a character in invoking string with another character
c) replace() method replaces all the characters in invoking string with another character
d) replace() method replaces last occurrence of a character in invoking string with another character

**Answer : A**

Explanation: replace() method replaces all occurrences of one character in invoking string with another character.

**Question 363:** What is the output of this program?

```
class output
{
   public static void main(String args[])
   {
     String c = "  Hello World  ";
     String s = c.trim();
     System.out.println("\""+s+"\"");
   }
}
```

a) ""Hello World""
b) ""Hello World"
c) "Hello World"
d) Hello world

**Answer : C**

Explanation: trim() method is used to remove leading and trailing whitespaces in a string.

**Question 364:** What is the output of this program?

```
class output
{
   public static void main(String args[])
   {
     String s1 = "one";
     String s2 = s1 + " two";
     System.out.println(s2);
   }
}
```

a) one
b) two
c) one two
d) compilation error

**Answer : C**

Explanation: None.

**Question 365:** What is the output of this program?

```
class output
{
   public static void main(String args[])
   {
     String s1 = "Hello";
     String s2 = s1.replace('l','w');
     System.out.println(s2);
   }
}
```

a) hello
b) helwo
c) hewlo
d) hewwo

**Answer : D**

Explanation: replace() method replaces all occurrences of one character in invoking string with another character. s1.replace('l','w') replaces every occurrence of 'l' in hello by 'w', giving hewwo.

**Question 366:** What is the output of this program?

```
class output
{
   public static void main(String args[])
   {
     String s1 = "Hello World";
     String s2 = s1.substring(0 , 4);
     System.out.println(s2);
   }
}
```

a) Hell
b) Hello
c) Worl
d) World

**Answer : A**

Explanation: substring(0,4) returns the character from 0 th position to 3 rd position.

**Question 367:** What is the output of this program?

```
class output
{
   public static void main(String args[])
   {          String s = "Hello World";
      int i = s.indexOf('o');
      int j = s.lastIndexOf('l');
      System.out.print(i + " " + j);

   }
}
```

a) 4 8
b) 5 9
c) 4 9
d) 5 8

**Answer : C**

Explanation: indexOf() method returns the index of first occurrence of the character where as lastIndexOf() returns the index of last occurrence of the character.

# StringBuffer Class

**Question 368:** Which class is used to create an object whose character sequence is mutable?

a) String()
b) StringBuffer()
c) String() & StringBuffer()
d) None of the mentioned

**Answer : B**

Explanation: StringBuffer represents growable and writable character sequence.

**Question 369:** Which of this method of class StringBuffer is used to concatenate the string representation to the end of invoking string?

a) concat()
b) append()
c) join()
d) concatenate()

**Answer : B**

Explanation: None.

**Question 370:** Which method of class StringBuffer is used to find the length of current character sequence?

a) length()
b) Length()
c) capacity()
d) Capacity()

**Answer : A**

Explanation: None.

**Question 371:** What is the string contained in s after following lines of code?

```
StringBuffer s new StringBuffer("Hello");
s.deleteCharAt(0);
```

a) Hell
b) ello
c) Hel
d) llo

**Answer : B**

Explanation: deleteCharAt() method deletes the character at the specified index location and returns the resulting StringBuffer object.

**Question 372:** Which of the following statement is correct?

a) reverse() method reverses all characters
b) reverseall() method reverses all characters
c) replace() method replaces first occurrence of a character in invoking string with another character
d) replace() method replaces last occurrence of a character in invoking string with another character

**Answer : A**

Explanation: reverse() method reverses all characters. It returns the reversed object on which it was called.

**Question 373:** What is the output of this program?

```
class output
{
   public static void main(String args[])
   {
      String a = "hello i love java";
      System.out.println(a.indexOf('e')+" "+a.indexOf('a')+" "+a.lastIndexOf('l')+" "+a.lastIndexOf('v'));
   }
}
```

a) 6 4 6 9
b) 5 4 5 9
c) 7 8 8 9
d) 1 14 8 15

**Answer : D**

Explanation: indexof(‘c’) and lastindexof(‘c’) are predefined function which are used to get the index of first and last occurrence of the character pointed by c in the given array.

**Question 374:** What is the output of this program?

```
class output
{
   public static void main(String args[])
   {
      StringBuffer c = new StringBuffer("Hello");
      c.delete(0,2);
      System.out.println(c);
   }
}
```

a) He
b) Hel
c) lo
d) llo

**Answer : D**

Explanation: delete(0,2) is used to delete the characters from 0 th position to 1 st position.

**Question 375:** What is the output of this program?

```
class output
{
   public static void main(String args[])
   {
       StringBuffer c = new StringBuffer("Hello");
       StringBuffer c1 = new StringBuffer(" World");
       c.append(c1);
       System.out.println(c);
   }
}
```

a) Hello
b) World
c) Helloworld
d) Hello World

**Answer : D**

Explanation: append() method of class StringBuffer is used to concatenate the string representation to the end of invoking string.

**Question 376:** What is the output of this program?

```
class output
{
   public static void main(String args[])
   {
      StringBuffer s1 = new StringBuffer("Hello");
      StringBuffer s2 = s1.reverse();
      System.out.println(s2);
   }
}
```

a) Hello
b) olleH
c) HelloolleH
d) olleHHello

**Answer : B**

Explanation: reverse() method reverses all characters. It returns the reversed object on which it was called.

**Question 377:** What is the output of this program?

```
class output
{
  class output
  {
    public static void main(String args[])
    {
      char c[]={'A', '1', 'b' , ' ' ,'a' , '0'};
      for (int i = 0; i < 5; ++i)
      {
          i++;
          if(Character.isDigit(c[i]))
            System.out.println(c[i]+" is a digit");
          if(Character.isWhitespace(c[i]))
            System.out.println(c[i]+" is a Whitespace character");
          if(Character.isUpperCase(c[i]))
            System.out.println(c[i]+" is an Upper case Letter");
          if(Character.isLowerCase(c[i]))
            System.out.println(c[i]+" is a lower case Letter");
          i++;
      }
    }
  }
```

a)
a is a lower case Letter
is White space character

b)
b is a lower case Letter
is White space character

c)
1 is a digit
a is a lower case Letter

d)
a is a lower case Letter
0 is a digit

**Answer : C**

Explanation: Character.isDigit(c[i]),Character.isUpperCase(c[i]),Character.isWhitespace(c[i]) are the function of library java.lang they are used to find whether the given character is of specified type or not. They return true or false i:e Boolean variable.

# Object & Math Class

**Question 378:** Which of these class is a superclass of all other classes?

a) Math
b) Process
c) System
d) Object

**Answer : D**

Explanation: The object class class is a superclass of all other classes.

**Question 379:** Which method of Object class can generate duplicate copy of the object on which it is called?

a) clone()
b) copy()
c) duplicate()
d) dito()

**Answer : A**

Explanation: None.

**Question 380:** What is the value of double consonant 'E' defined in Math class?

a) approximately 3
b) approximately 3.14
c) approximately 2.72
d) approximately 0

**Answer : C**

Explanation: None.

**Question 381:** Which of these classes contains only floating point functions?

a) Math
b) Process
c) System
d) Object

**Answer : A**

Explanation: Math class contains all the floating point functions that are used for geometry, trigonometry, as well as several general purpose methods. Example : sin(), cos(), exp(), sqrt() etc.

**Question 382:** What is the value of "d" after this line of code has been executed?

```
double d = Math.round ( 2.5 + Math.random() );
```

a) 2
b) 3
c) 4
d) 2.5

**Answer : B**

Explanation: The Math.random() method returns a number greater than or equal to 0 and less than 1. so 2.5 will be greater than or equal to 2.5 and less than 3.5, we can be sure that Math.round() will round that number to 3.

**Question 383:** What is the output of this program?

```
class Output
{
    public static void main(String args[])
    {
       int x = 3.14;
       int y = (int) Math.abs(x);
       System.out.print(y);
    }
}
```

a) 0
b) 3
c) 3.0
d) 3.1

**Answer : B**

Explanation: None.



**Question 384:** What is the output of this program?

```
class Output
{
    public static void main(String args[])
    {
       double x = 3.1;
       double y = 4.5;
       double z = Math.max( x, y );
       System.out.print(z);
    }
}
```

a) true
b) 4
c) 3.1
d) 4.5

**Answer : D**

Explanation: None.

**Question 385:** What is the output of this program?

```
class Output
{
    public static void main(String args[])
   {
      double x = 2.0;
      double y = 3.0;
      double z = Math.pow( x, y );
      System.out.print(z);
   }
}
```

a) 9
b) 8
c) 8.0
d) 9.0

**Answer : C**

Explanation: Math.pow(x, y) methods returns value of y to the power x, i:e x ^ y, 2.0 ^ 3.0 = 8.0.

# Java.io Byte & Character Streams

**Question 386:** Which of these classes is used for input and output operation when working with bytes?

a) InputStream
b) Reader
c) Writer
d) All of the mentioned

**Answer : A**

Explanation: InputStream & OutputStream are designed for byte stream. Reader and writer are designed for character stream.

**Question 387:** Which of these classes is used to read and write bytes in a file?

a) FileReader
b) FileWriter
c) FileInputStream
d) InputStreamReader

**Answer : C**

Explanation: None.

**Question 388:** Which method of InputStream is used to read integer representation of next available byte input?

a) read()
b) scanf()
c) get()
d) getInteger()

**Answer : A**

Explanation: None.

**Question 389:** Which of these is a method to clear all the data present in output buffers?

a) clear()
b) flush()
c) fflush()
d) close()

**Answer : B**

Explanation: None.

**Question 390:** Which of these method(s) is/are used for writing bytes to an outputstream?

a) put()
b) print() and write()
c) printf()
d) write() and read()

**Answer : B**

Explanation: write() and print() are the two methods of OutputStream that are used for printing the byte data.

**Question 391:** What is the output of this program?

```
import java.io.*;
class filesinputoutput
{
   public static void main(String args[])
   {
      InputStream obj = new FileInputStream("inputoutput.java");
      System.out.print(obj.available());
   }
}
```

**Note**: inputoutput.java is stored in the disk

a) true
b) false
c) prints number of bytes in file
d) prints number of characters in the file

**Answer : C**

Explanation: obj.available() returns the number of bytes.



**Question 392:** What is the output of this program?

```
import java.io.*;
public class filesinputoutput
{
    public static void main(String[] args)
    {
      String obj  = "abc";
      byte b[] = obj.getBytes();
      ByteArrayInputStream obj1 = new ByteArrayInputStream(b);
```

```
                    for (int i = 0; i < 2; ++ i)
                    {
                       int c;
                       while ((c = obj1.read()) != -1)
                       {
                                  if(i == 0)
                                     System.out.print((char)c);
                       }
                    }
                 }
              }
```

a) abc
b) ABC
c) ab
d) AB

**Answer : A**

Explanation: None.

**Question 393:** What is the output of this program?

```
import java.io.*;
public class filesinputoutput
{
   public static void main(String[] args)
   {
        String obj  = "abc";
        byte b[] = obj.getBytes();
        ByteArrayInputStream obj1 = new ByteArrayInputStream(b);
        for (int i = 0; i < 2; ++ i)
        {
           int c;
           while ((c = obj1.read()) != -1)
           {
                    if (i == 0)
                      System.out.print(Character.toUpperCase((char)c));
           }
        }
   }
}
```

a) abc
b) ABC
c) ab
d) AB

**Answer : B**

Explanation: None.

**Question 394:** What is the output of this program?

```
import java.io.*;
public class filesinputoutput
{
    public static void main(String[] args)
    {
       String obj  = "abc";
       byte b[] = obj.getBytes();
       ByteArrayInputStream obj1 = new ByteArrayInputStream(b);
       for (int i = 0; i < 2; ++ i)
       {
          int c;
          while ((c = obj1.read()) != -1)
          {
                  if (i == 0)
                  {
                      System.out.print(Character.toUpperCase((char)c));
                      obj2.write(1);
                  }
           }
           System.out.print(obj2);
       }
    }
}
```

a) AaBaCa
b) ABCaaa
c) AaaBaaCaa
d) AaBaaCaaa

**Answer : D**

Explanation: None.

**Question 395:** Which of this class is used to read characters in a file?

a) FileReader
b) FileWriter
c) FileInputStream
d) InputStreamReader

**Answer : A**

Explanation: None.

**Question 396:** Which method of FileReader class is used to read characters from a file?

a) read()
b) scanf()
c) get()
d) getInteger()

**Answer : A**

Explanation: None.

# Java's Built in Exceptions

**Question 397:** Which of these exceptions handles the situations when an illegal argument is used to invoke a method?

a) IllegalException
b) Argument Exception
c) IllegalArgumentException
d) IllegalMethodArgumentExcepetion

**Answer : C**

Explanation: None.

**Question 398:** Which of these exceptions will be thrown if we declare an array with negative size?

a) IllegalArrayException
b) IllegalArraySizeExeption
c) NegativeArrayException
d) NegativeArraySizeExeption

**Answer : D**

Explanation: Array size must always be positive if we declare an array with negative size then built in exception "NegativeArraySizeException" is thrown by the java's run time system.

**Question 399:** Which of these packages contain all the Java's built in exceptions?

a) java.io
b) java.util
c) java.lang
d) java.net

**Answer : C**

Explanation: None.



**Question 400:** Which of these exceptions will be thrown if we use null reference for an arithmetic operation?

a) ArithmeticException
b) NullPointerException
c) IllegalAccessException
d) IllegalOperationException

**Answer : B**

Explanation: If we use null reference anywhere in the code where the value stored in that reference is used then NullPointerException occurs.

**Question 401:** Which of this class is used to create user defined exception?

a) java.lang
b) Exception
c) RunTime
d) OwnException

**Answer : B**

Explanation: Exception class contains all the methods necessary for defining an exception. The class contains the Throwable class.

**Question 402:** What is the output of this program?

```
class exception_handling
{
   public static void main(String args[])
   {
      try
      {
         int a[] = {1, 2, 3, 4, 5};
         for (int i = 0; i < 7; ++i)
            System.out.print(a[i]);
      }
      catch(ArrayIndexOutOfBoundsException e)
      {
     System.out.print("0");
      }
   }
}
```

a) 12345
b) 123450
c) 1234500
d) Compilation Error



**Answer : B**

Explanation: When array index goes out of bound then ArrayIndexOutOfBoundsException exception is thrown by the system.

**Question 403:** What is the output of this program?

```
class exception_handling
{
   public static void main(String args[])
   {
      try
      {
         int a[] = {1, 2,3 , 4, 5};
         for (int i = 0; i < 5; ++i)
               System.out.print(a[i]);
         int x = 1/0;
      }
      catch(ArrayIndexOutOfBoundsException e)
      {
          System.out.print("A");
      }
      catch(ArithmeticException e)
      {
         System.out.print("B");
      }
   }
}
```

a) 12345
b) 12345A
c) 12345B
d) Compilation Error

**Answer : C**

Explanation: There can be more than one catch of a single try block. Here Arithmetic exception occurs instead of Array index out of bound exception hence B is printed after 12345



**Question 404:** What is the output of this program?

```
class exception_handling
{
      static void throwexception() throws ArithmeticException
      {
         System.out.print("0");
         throw new ArithmeticException ("Exception");
      }
      public static void main(String args[])
      {
      try
      {
         throwexception();
      }
```

```
                catch (ArithmeticException e)
                {
                      System.out.println("A");
                }
            }
        }
```

a) A
b) 0
c) 0A
d) Exception

**Answer : C**

Explanation: None.

**Question 405:** What is the output of this program?

```
class exception_handling
{
   public static void main(String args[])
   {
      try
      {
         int a = 1;
         int b = 10 / a;
         try
         {
            if (a == 1)
               a = a / a - a;
            if (a == 2)
            {
               int c[] = {1};
               c[8] = 9;
            }
         finally
         {
            System.out.print("A");
         }

      }
      catch (NullPointerException e)
      {
            System.out.println("B");
      }
   }
}
```

a) A
b) B
c) AB
d) BA

**Answer : A**

Explanation: The inner try block does not have a catch which can tackle ArrayIndexOutOfBoundException hence finally is executed which prints 'A' the outer try block does have catch for NullPointerException exception but no such exception occurs in it hence its catch is never executed and only 'A' is printed.

**Question 406:** What is the output of this program?

```
class exception_handling
{
   public static void main(String args[])
   {
      try
      {
         int a = args.length;
         int b = 10 / a;
         System.out.print(a);
         try
         {
            if (a == 1)
               a = a / a - a;
            if (a == 2)
            {
               int c = {1};
               c[8] = 9;
            }
         }
         catch (ArrayIndexOutOfBoundException e)
         {
            System.out.println("TypeA");
         }
         catch (ArithmeticException e)
         {
            System.out.println("TypeB");
         }
      }
   }
```

a) TypeA
b) TypeB
c) 0TypeA
d) 0TypeB

**Answer : D**

Explanation: Execution command line is "$ java exception_ handling one two" hence there are two input making args.length = 2, hence "c[8] = 9" in second try block is executing which throws ArrayIndexOutOfBoundException which is caught by catch of nested try block. Hence 0TypeB is printed

# Java.lang – Rounding Functions

**Question 407:** Which of this class provides various types of rounding functions?

a) Math
b) Process
c) System
d) Object

**Answer : A**

Explanation: None.

**Question 408:** Which of these methods return a smallest whole number greater than or equal to variable X?

a) double ceil(double X)
b) double floor(double X)
c) double max(double X)
d) double min(double X)

**Answer : A**

Explanation: ceil(double X) returns the smallest whole number greater than or equal to variable X.

**Question 409:** Which of these method returns a largest whole number less than or equal to variable X?

a) double ceil(double X)
b) double floor(double X)
c) double max(double X)
d) double min(double X)

**Answer : B**

Explanation: double floor(double X) returns a largest whole number less than or equal to variable X.

**Question 410:** Which function return absolute value of a variable?

a) abs()
b) absolute()
c) absolutevariable()
d) none of the mentioned

**Answer : A**

Explanation: abs() returns the absolute value of a variable.

**Question 411:** What is the output of this program?

```
class Output
{
    public static void main(String args[])
    {
        double x = 3.14;
        int y = (int) Math.ceil(x);
        System.out.print(y);
    }
}
```

a) 0
b) 3
c) 3.0
d) 4

**Answer : D**

Explanation: ceil(double X) returns the smallest whole number greater than or equal to variable x.

**Question 412:** What is the output of this program?

```
class Output
{
    public static void main(String args[])
    {
        double x = 3.14;
        int y = (int) Math.floor(x);
        System.out.print(y);
    }
}
```

a) 0
b) 3
c) 3.0
d) 4

**Answer : B**

Explanation: double floor(double X) returns a largest whole number less than or equal to variable X. Here the smallest whole number less than 3.14 is 3.

# ThreadGroup class & Runnable Interface

**Question 413:** Which of the interface contains all the methods used for handling thread related operations in Java?

a) Runnable interface
b) Math interface
c) System interface
d) ThreadHandling interface

**Answer : A**

Explanation: Runnable interface defines all the methods for handling thread operations in Java.

**Question 414:** Which class is used to make a thread?

a) String
b) System
c) Thread
d) Runnable

**Answer : C**

Explanation: Thread class is used to make threads in java, Thread encapsulates a thread of execution. To create a new thread the program will either extend Thread or implement the Runnable interface.

**Question 415:** Which of these methods of Thread class is used to suspend a thread for a period of time?

a) sleep()
b) terminate()
c) suspend()
d) stop()

**Answer : A**

Explanation: None.



**Question 416:** What is the output of this program?

```
class newthread implements Runnable
{
    Thread t1,t2;
    newthread()
    {
       t1 = new Thread(this,"Thread_1");
       t2 = new Thread(this,"Thread_2");
       t1.start();
       t2.start();
    }
```

```
        public void run()
        {
            t2.setPriority(Thread.MAX_PRIORITY);
            System.out.print(t1.equals(t2));
        }
    }
    class multithreaded_programing
    {
       public static void main(String args[])
       {
          new newthread();
       }
    }
```

a) true
b) false
c) truetrue
d) falsefalse

**Answer : D**

Explanation: Threads t1 & t2 are created by class newthread that is implementing runnable interface, hence both the threads are provided their own run() method specifying the actions to be taken. When constructor of newthread class is called first the run() method of t1 executes than the run method of t2 printing 2 times "false" as both the threads are not equal one is having different priority than other, hence falsefalse is printed.

**Question 417:** What is the output of this program?

```
class newthread implements Runnable
{
    Thread t
    newthread()
    {
     t = new Thread(this,"New Thread");
     t.start();
    }
    public void run()
    {
     t.setPriority(Thread.MAX_PRIORITY);
     System.out.println(t);
    }
}
```

```
class multithreaded_programing
{
   public static void main(String args[])
   {
      new newthread();
   }
}
```

a) Thread[New Thread,0,main].
b) Thread[New Thread,1,main].
c) Thread[New Thread,5,main].
d) Thread[New Thread,10,main].

**Answer : B**

Explanation: Thread t has been made with default priority value 5 but in run method the priority has been explicitly changed to MAX_PRIORITY of class thread, that is 10 by code 't.setPriority(Thread.MAX_PRIORITY);' using the setPriority function of thread t.

**Question 418:** What is the output of this program?

```
class newthread implements Runnable
{
     Thread t;
     newthread()
     {
        t = new Thread(this,"My Thread");
        t.start();
     }
}
class multithreaded_programing
{
   public static void main(String args[])
   {
      new newthread();
   }
}
```

a) My Thread
b) Thread[My Thread,5,main].
c) Compilation Error
d) Runtime Error

**Answer : C**

Explanation: Thread t has been made by using Runnable interface, hence it is necessary to use inherited abstract method run() method to specify instructions to be implemented on the thread, since no run() method is used it gives a compilation error.

**Question 419:** What is the output of this program?

```
class newthread implements Runnable
{
    Thread t;
    newthread()
    {
        t = new Thread(this,"My Thread");
        t.start();
    }
    public void run()
    {
        System.out.println(t.getName());
    }
}
class multithreaded_programing
{
    public static void main(String args[])
    {
        new newthread();
    }
}
```

a) My Thread
b) Thread[My Thread,5,main]
c) Compilation Error
d) Runtime Error

**Answer : A**

Explanation: None.

**Question 420:** What is the output of this program?

```
class newthread implements Runnable
{
    Thread t;
    newthread()
    {
        t = new Thread(this,"My Thread");
        t.start();
    }
    public void run()
    {
        System.out.println(t);
    }
}
```

```
class multithreaded_programing
{
   public static void main(String args[])
   {
      new newthread();
   }
}
```

a) My Thread
b) Thread[My Thread,5,main].
c) Compilation Error
d) Runtime Error

**Answer : B**

Explanation: None.

# Java.util – ArrayList Class

**Question 421:** Which standard collection class implements a dynamic array?

a) AbstractList
b) LinkedList
c) ArrayList
d) ArraySet

**Answer : C**

Explanation: ArrayList class implements a dynamic array by extending AbstractList class.

**Question 422:** Which method can be used to increase the capacity of ArrayList object manually?

a) Capacity()
b) increaseCapacity()
c) increasecapacity()
d) ensureCapacity()

**Answer : D**

Explanation: When we add an element, the capacity of ArrayList object increases automatically, but we can increase it manually to specified length x by using function ensureCapacity(x);

**Question 423:** Which method of ArrayList class is used to obtain present size of an object?

a) size()
b) length()
c) index()
d) capacity()

**Answer : A**

Explanation: None.

**Question 424:** Which method can be used to obtain a static array from an ArrayList object?

a) Array()
b) covertArray()
c) toArray()
d) covertoArray()

**Answer : C**

Explanation: None.

**Question 425:** Which method is used to reduce the capacity of an ArrayList object?

a) trim()
b) trimSize()
c) trimTosize()
d) trimToSize()

**Answer : D**

Explanation: trimTosize() is used to reduce the size of the array that underlines an ArrayList object.

**Question 426:** What is the output of this program?

```
import java.util.*;
class Arraylist
{
   public static void main(String args[])
   {
      ArrayList obj = new ArrayList();
      obj.add("A");
      obj.add("B");
      obj.add("C");
      obj.add(1, "D");
      System.out.println(obj);
   }
}
```

a) [A, B, C, D]
b) [A, D, B, C]
c) [A, D, C]
d) [A, B, C]

**Answer : B**

Explanation: obj is an object of class ArrayList hence it is an dynamic array which can increase and decrease its size. obj.add("X") adds to the array element X and obj.add(1,"X") adds element x at index position 1 in the list, Hence obj.add(1,"D") stores D at index position 1 of obj and shifts the previous value stored at that position by 1.

**Question 427:** What is the output of this program?

```
import java.util.*;
class Output
{
   public static void main(String args[])
   {
      ArrayList obj = new ArrayList();
      obj.add("A");
      obj.add(0, "B");
      System.out.println(obj.size());
   }
}
```

a) 0
b) 1
c) 2
d) Any Garbage Value

**Answer : C**

Explanation: None.

**Question 428:** What is the output of this program?

```
import java.util.*;
class Output
{
   public static void main(String args[])
   {
      ArrayList obj = new ArrayList();
      obj.add("A");
      obj.ensureCapacity(3);
      System.out.println(obj.size());
   }
}
```

a) 1
b) 2
c) 3
d) 4

**Answer : A**

Explanation: Although obj.ensureCapacity(3); has manually increased the capacity of obj to 3 but the value is stored only at index 0, therefore obj.size() returns the total number of elements stored in the obj i:e 1, it has nothing to do with ensureCapacity().

**Question 429:** What is the output of this program?

```
class Output
{
   public static void main(String args[])
   {
      ArrayList obj = new ArrayList();
      obj.add("A");
      obj.add("D");
      obj.ensureCapacity(3);
      obj.trimToSize();
      System.out.println(obj.size());
    }
}
```

a) 1
b) 2
c) 3
d) 4

**Answer : B**

Explanation: trimTosize() is used to reduce the size of the array that underlines an ArrayList object.

# Java.util – Array Class

**Question 430:** Which of these standard collection classes implements all the standard functions on list data structure?

a) Array
b) LinkedList
c) HashSet
d) AbstractSet

**Answer : A**

Explanation: None.

**Question 431:** Which of this method is used to make all elements of an equal to specified value?

a) add()
b) fill()
c) all()
d) set()

**Answer : B**

Explanation: fill() method assigns a value to all the elements in an array, in other words, it fills the array with specified value.

**Question 432:** Which method of Array class is used sort an array or its subset?

a) binarysort()
b) bubblesort()
c) sort()
d) insert()

**Answer : C**

Explanation: None.

**Question 433:** Which method can be used to search an element in a list?

a) find()
b) sort()
c) get()
d) binaryserach()

**Answer : D**

Explanation: binaryserach() method uses binary search to find a specified value. This method must be applied to sorted arrays.

**Question 434:** What is the output of this program?

```
import java.util.*;
class Arraylist
{
   public static void main(String args[])
   {
      ArrayList obj1 = new ArrayList();
      ArrayList obj2 = new ArrayList();
      obj1.add("A");
      obj1.add("B");
      obj2.add("A");
      obj2.add(1, "B");
      System.out.println(obj1.equals(obj2));
   }
}
```

a) 0
b) 1
c) true
d) false

**Answer : C**

Explanation: obj1 and obj2 are an object of class ArrayList hence it is a dynamic array which can increase and decrease its size. obj.add("X") adds to the array element X and obj.add(1,"X") adds element x at index position 1 in the list, Both the objects obj1 and obj2 contain same elements i:e A & B thus obj1.equals(obj2) method returns true.

**Question 435:** What is the output of this program?

```
import java.util.*;
class Array
{
   public static void main(String args[])
   {
      int array[] = new int [5];
      for (int i = 5; i > 0; i--)
         array[5 - i] = i;
      Arrays.sort(array);
      for (int i = 0; i < 5; ++i)
                  System.out.print(array[i]);;
   }
}
```

a) 12345
b) 54321
c) 1234
d) 5432

**Answer : A**

Explanation: Arrays.sort(array) method sorts the array into 1,2,3,4,5.

**Question 436:** What is the output of this program?

```
import java.util.*;
class Array
{
   public static void main(String args[])
   {
      int array[] = new int [5];
      for (int i = 5; i > 0; i--)
         array[5 - i] = i;
      Arrays.sort(array);
      System.out.print(Arrays.binarySearch(array, 4));
   }
}
```

a) 2
b) 3
c) 4
d) 5

**Answer : B**

Explanation: None.

# Exceptional Handling

**Question 437:** When does Exceptions in Java arises in code sequence?

a) Run Time
b) Compilation Time
c) Can Occur Any Time
d) None of the mentioned

**Answer : A**

Explanation: Exceptions in Java are run-time errors.

**Question 438:** Which of these keywords is not a part of exception handling?

a) try
b) finally
c) thrown
d) catch

**Answer : C**

Explanation: Exceptional handling is managed via 5 keywords – try, catch, throws, throw and finally.

**Question 439:** Which keyword must be used to monitor for exceptions?

a) try
b) finally
c) throw
d) catch

**Answer : A**

Explanation: None.

**Question 440:** Which keyword must be used to handle the exception thrown by try block in some rational manner?

a) try
b) finally
c) throw
d) catch

**Answer : D**

Explanation: If an exception occurs within the try block, it is thrown and cached by catch block for processing.

**Question 441:** Which keyword is used to manually throw an exception?

a) try
b) finally
c) throw
d) catch

**Answer : C**

Explanation: None.

**Question 442:** What is the output of this program?

```
class exception_handling
{
   public static void main(String args[])
   {
      try
      {
          System.out.print("Hello" + " " + 1 / 0);
      }
      catch(ArithmeticException e)
      {
          System.out.print("World");
      }
   }
}
```

a) Hello
b) World
c) HelloWorld
d) Hello World

**Answer : B**

Explanation: System.out.print() function first converts the whole parameters into a string and then prints, before "Hello" goes to output stream 1 / 0 error is encountered which is cached by catch block printing just "World".

**Question 443:** What is the output of this program?

```
class exception_handling
{
   public static void main(String args[])
   {
      try
      {
         int a, b;
         b = 0;
         a = 5 / b;
         System.out.print("A");
      }
      catch(ArithmeticException e)
      {
          System.out.print("B");
      }
   }
}
```

a) A
b) B
c) Compilation Error
d) Runtime Error

**Answer : B**

Explanation: None.

**Question 444:** What is the output of this program?

```
class exception_handling
{
   public static void main(String args[])
   {
      try
      {
         int a, b;
         b = 0;
         a = 5 / b;
         System.out.print("A");
      }
      catch(ArithmeticException e)
      {
          System.out.print("B");
      }
      finally
      {
          System.out.print("C");
      }
   }
}
```

a) A
b) B
c) AC
d) BC

**Answer : D**

Explanation: finally keyword is used to execute the code before try and catch block end.

**Question 445:** What is the output of this program?

```
class exception_handling
{
   public static void main(String args[])
   {
      try
      {
         int i, sum;
         sum = 10;
         for (i = -1; i < 3 ;++i)
            sum = (sum / i);
      }
      catch(ArithmeticException e)
      {
           System.out.print("0");
      }
      System.out.print(sum);
   }
}
```

a) 0
b) 05
c) Compilation Error
d) Runtime Error

**Answer : C**

Explanation: Value of variable sum is printed outside of try block, sum is declared only in try block, outside try block it is undefined.



**Question 446:** Which of the following class can catch all exceptions?

a) RuntimeException
b) Error
c) Exception
d) ParentException

**Answer : C**

Explanation: None.

**Question 447:** Which of the following is a super class of all exception type classes?

a) Catchable
b) RuntimeExceptions
c) String
d) Throwable

**Answer : D**

Explanation: Throwable is built in class and all exception types are subclass of this class. It is the super class of all exceptions.

**Question 448:** Which of the following operator is used to generate instance of an exception which can be thrown using throw?

a) thrown
b) alloc
c) malloc
d) new

**Answer : D**

Explanation: new operator is used to create instance of an exception. Exceptions may have parameter as a String or have no parameter.

**Question 449:** Which of the following keyword is used by calling function to handle exception thrown by called function?

a) throws
b) throw
c) try
d) catch

**Answer : A**

Explanation: A method specifies behaviour of being capable of causing exception. Throws clause in the method declaration guards caller of the method from exception.

**Question 450:** Which of the following handles the exception when a catch is not used?

a) finally
b) throw handler
c) default handler
d) java run time system

**Answer : C**

Explanation: Default handler is used to handle all the exceptions if catch is not used to handle exception. Finally is called in any case.

**Question 451:** Which part of code gets executed whether exception is caught or not?

a) finally
b) try
c) catch
d) throw

**Answer : A**

Explanation: Finally block of the code gets executed regardless exception is caught or not. File close, database connection close, etc are usually done in finally.

**Question 452:** At runtime, error is recoverable.

a) True
b) False

**Answer : B**

Explanation: Error is not recoverable at runtime. The control is lost from the application.

**Question 453:** Which class is related to all the exceptions that cannot be caught?

a) Error
b) Exception
c) RuntimeExecption
d) All of the mentioned

**Answer : A**

Explanation: Error class is related to java run time error that can't be caught usually, RuntimeExecption is subclass of Exception class which contains all the exceptions that can be caught.

**Question 454:** What exception thrown by parseInt() method?

a) ArithmeticException
b) ClassNotFoundException
c) NullPointerException
d) NumberFormatException

**Answer : D**

Explanation: parseInt() method parses input into integer. The exception thrown by this method is NumberFormatException.

**Question 455:** What is the output of this program?

```
class exception_handling
{
   public static void main(String args[])
   {
      try
      {
          System.out.print("Hello" + " " + 1 / 0);
      }
```

```
                finally
                {
                    System.out.print("World");
                }
            }
        }
```

a) Hello
b) World
c) Compilation Error
d) It prints information about Exception then World

**Answer : D**

Explanation: None.

**Question 456:** What is the output of this program?

```
class exception_handling
{
    public static void main(String args[])
    {
        try
        {
            int i, sum;
            sum = 10;
            for (i = -1; i < 3 ;++i)
            {
                sum = (sum / i);
                System.out.print(i);
            }
        }
        catch(ArithmeticException e)
        {
            System.out.print("0");
        }
    }
}
```

a) -1
b) 0
c) -10
d) -101

**Answer : C**

Explanation: For the 1st iteration -1 is displayed. The 2nd exception is caught in catch block and 0 is displayed.

**Question 457:** Which keyword is used to generate an exception explicitly?

a) try
b) finally
c) throw
d) catch

**Answer : C**

Explanation: None.

**Question 458:** What is the output of this program?

```
class exception_handling
{
   public static void main(String args[])
   {
      try
      {
         int a = args.length;
         int b = 10 / a;
         System.out.print(a);
         try
         {
            if (a == 1)
               a = a / a - a;
            if (a == 2)
            {
               int []c = {1};
               c[8] = 9;
            }
         }
         catch (ArrayIndexOutOfBoundException e)
         {
            System.out.println("TypeA");
         }
         catch (ArithmeticException e)
         {
            System.out.println("TypeB");
         }
      }
   }
}
```

a) TypeA
b) TypeB
c) Compile Time Error
d) 0TypeB

**Answer : C**

Explanation: Because we can't go beyond array limit

**Question 459:** What is the output of this program?

```
class exception_handling
{
   public static void main(String args[])
   {
      try
      {
         System.out.print("A");
         throw new NullPointerException ("Hello");
      }
      catch(ArithmeticException e)
      {
         System.out.print("B");
      }
   }
}
```

a) A
b) B
c) Hello
d) Runtime Exception

**Answer : D**

Explanation: None.

**Question 460:** What is the output of this program?

```
public class San
 {
   public static void main(String[] args)
   {
      try
      {
      return;
      }
      finally
      {
         System.out.println( "Finally" );
      }
   }
}
```

a) Finally
b) Compilation fails
c) The code runs with no output
d) An exception is thrown at runtime

**Answer : A**

Explanation: Because finally will execute always.

**Question 461:** What is the output of this program?

```
public class San
{
   public static void main(String args[])
   {
      try
      {
         System.out.print("Hello world ");
      }
      finally
      {
         System.out.println("Finally executing ");
      }
   }
}
```

a) The program will not compile because no exceptions are specified
b) The program will not compile because no catch clauses are specified
c) Hello world
d) Hello world Finally executing

**Answer : D**

Explanation: None.

**Question 462:** A single try block must be followed by which of these?

a) finally
b) catch
c) finally & catch
d) none of the mentioned

**Answer : C**

Explanation: try block can be followed by any of finally or catch block, try block checks for exceptions and work is performed by finally and catch block as per the exception.

**Question 463:** Which of these exceptions handles the divide by zero error?

a) ArithmeticException
b) MathException
c) IllegalAccessException
d) IllegarException

**Answer : A**

Explanation: None.

**Question 464:** Which of these exceptions will occur if we try to access the index of an array beyond its length?

a) ArithmeticException
b) ArrayException
c) ArrayIndexException
d) ArrayIndexOutOfBoundsException

**Answer : D**

Explanation: ArrayIndexOutOfBoundsException is a built in exception that is caused when we try to access an index location which is beyond the length of an array.

**Question 465:** What is the output of this program?

```
class exception_handling
{
   public static void main(String args[])
   {
      try
      {
         int a = args.length;
         int b = 10 / a;
         System.out.print(a);
      }
      catch (ArithmeticException e)
      {
            System.out.println("1");
      }
   }
}
```

Note : Execution command line : $ java exception_handling

a) 0
b) 1
c) Compilation Error
d) Runtime Error

**Answer : B**

Explanation: None.

**Question 466:** What is the output of this program?

```
class exception_handling
{
   public static void main(String args[])
   {
      try
      {
         throw new NullPointerException ("Hello");
      }
      catch(ArithmeticException e)
      {
          System.out.print("B");
      }
   }
}
```

a) A
b) B
c) Compilation Error
d) Runtime Error

**Answer : D**

Explanation: Try block is throwing NullPointerException but the catch block is used to counter Arithmetic Exception. Hence NullPointerException occurs since no catch is there which can handle it, runtime error occurs.

**Question 467:** What is the output of this program?

```
class exception_handling
{
   public static void main(String args[])
   {
      try
      {
         int a = 1;
         int b = 10 / a;
         try
         {
            if (a == 1)
               a = a / a - a;
            if (a == 2)
            {
               int c[] = {1};
               c[8] = 9;
            }
         }
```

```
                finally
                {
                    System.out.print("A");
                }
            }
            catch (Exception e)
            {
                System.out.println("B");
            }
        }
    }
```

a) A
b) B
c) AB
d) BA

**Answer : A**

Explanation: The inner try block does not have a catch which can tackle ArrayIndexOutOfBoundException hence finally is executed which prints 'A' the outer try block does have catch for ArrayIndexOutOfBoundException exception but no such exception occurs in it hence its catch is never executed and only 'A' is printed.



**Question 468:** What is the output of this program?

```
class exception_handling
{
    public static void main(String args[])
    {
        try
        {
            int a = args.length;
            int b = 10 / a;
            System.out.print(a);
            try
            {
                if (a == 1)
                    a = a / a - a;
                if (a == 2)
                {
                    int []c = {1};
                    c[8] = 9;
                }
            }
            catch (ArrayIndexOutOfBoundException e)
            {
                System.out.println("TypeA");
            }
```

```
                catch (ArithmeticException e)
                {
                    System.out.println("TypeB");
                }
            }
        }
```

Note: Execution command line: $ java exception_handling one two

a) TypeA
b) TypeB
c) Compilation Error
d) Runtime Error

**Answer : C**

Explanation: try without catch or finally.

**Question 469:** What is the use of try & catch?

a) It allows us to manually handle the exception
b) It allows to fix errors
c) It prevents automatic terminating of the program in cases when an exception occurs
d) All of the mentioned

**Answer : D**

Explanation: None.

**Question 470:** Which of these keywords are used for the block to be examined for exceptions?

a) try
b) catch
c) throw
d) check

**Answer : A**

Explanation: try is used for the block that needs to checked for exception.

**Question 471:** Which of these keywords are used for the block to handle the exceptions generated by try block?

a) try
b) catch
c) throw
d) check

**Answer : B**

Explanation: None.

**Question 472:** Which of these statements is incorrect?

a) try block need not to be followed by catch block
b) try block can be followed by finally block instead of catch block
c) try can be followed by both catch and finally block
d) try need not to be followed by anything

**Answer : D**

Explanation: try must be followed by either catch or finally block.

**Question 473:** What is the output of this program?

```
class Output
{
   public static void main(String args[])
   {
     try
     {
        int a = 0;
        int b = 5;
        int c = b / a;
        System.out.print("Hello");
     }
     catch(Exception e)
     {
        System.out.print("World");
     }
   }
}
```

a) Hello
b) World
c) HelloWOrld
d) Compilation Error

**Answer : B**

Explanation: None.



**Question 474:** What is the output of this program?

```
class Output
{
   public static void main(String args[])
   {
     try
     {
        int a = 0;
        int b = 5;
        int c = a / b;
        System.out.print("Hello");
     }
```

```
                catch(Exception e)
                {
                   System.out.print("World");
                }
              }
            }
```

a) Hello
b) World
c) HelloWOrld
d) Compilation Error

**Answer : A**

Explanation: None.

**Question 475:** What is the output of this program?

```
class Output
{
   public static void main(String args[])
   {
      try
      {
         int a = 0;
         int b = 5;
         int c = b / a;
         System.out.print("Hello");
      }
   }
}
```

a) Hello
b) World
c) HelloWOrld
d) Compilation Error

**Answer : D**

Explanation: try must be followed by either catch or finally.

**Question 476:** What is the output of this program?

```
class Output
{
   public static void main(String args[])
   {
      try
      {
         int a = 0;
         int b = 5;
         int c = a / b;
         System.out.print("Hello");
      }
      finally
      {
         System.out.print("World");
      }
   }
}
```

a) Hello
b) World
c) HelloWOrld
d) Compilation Error

**Answer : C**

Explanation: finally block is always executed after try block, no matter exception is found or not.

**Question 477:** What is the output of this program?

```
class Output
{
   public static void main(String args[])
   {
      try
      {
         int a = 0;
         int b = 5;
         int c = b / a;
         System.out.print("Hello");
      }
      catch(Exception e)
      {
         System.out.print("World");
      }
```

```
            finally
            {
                System.out.print("World");
            }
        }
    }
```

a) Hello
b) World
c) HelloWOrld
d) WorldWorld

**Answer : D**

Explanation: finally block is always executed after tryblock, no matter exception is found or not. catch block is executed only when exception is found. Here divide by zero exception is found hence both catch and finally are executed.

**Question 478:** Which method return description of an exception?

a) getException()
b) getMessage()
c) obtainDescription()
d) obtainException()

**Answer : B**

Explanation: getMessage() returns a description of the exception.

**Question 479:** Which of these methods is used to print stack trace?

a) obtainStackTrace()
b) printStackTrace()
c) getStackTrace()
d) displayStackTrace()

**Answer : B**

Explanation: None.

**Question 480:** What is the output of this program?

```
class Myexception extends Exception
{
    int detail;
    Myexception(int a)
    {
        detail = a;
    }
    public String toString()
    {
            return "detail";
    }
}
```

```
class Output
{
   static void compute (int a) throws Myexception
   {
        throw new Myexception(a);
   }
   public static void main(String args[])
   {
      try
      {
         compute(3);
      }
     catch(Myexception e)
     {
        System.out.print("Exception");
     }
   }
}
```

a) 3
b) Exception
c) Runtime Error
d) Compilation Error

**Answer : B**

Explanation: Myexception is self defined exception.



**Question 481:** What is the output of this program?

```
class Myexception extends Exception
{
   int detail;
   Myexception(int a)
   {
        detail = a;
    }
   public String toString()
   {
        return "detail";
   }
}
class Output
{
   static void compute (int a) throws Myexception
   {
        throw new Myexception(a);
   }
```

```
public static void main(String args[])
{
   try
   {
      compute(3);
   }
   catch(DevideByZeroException e)
   {
      System.out.print("Exception");
   }
  }
}
```

a) 3
b) Exception
c) Runtime Error
d) Compilation Error

**Answer : C**

Explanation: Mexception is self defined exception, we are generating Myexception but catching DevideByZeroException which causes error.

**Question 482:** What is the output of this program?

```
class exception_handling
{
   public static void main(String args[])
   {
      try
      {
         throw new NullPointerException ("Hello");
         System.out.print("A");
      }
      catch(ArithmeticException e)
      {
            System.out.print("B");
      }
   }
}
```

a) A
b) B
c) Compilation Error
d) Runtime Error

**Answer : D**

Explanation: try block is throwing NullPointerException but the catch block is used to counter Arithmetic Exception. Hence NullPointerException occurs since no catch is there which can handle it, runtime error occurs.

**Question 483:** What is the output of this program?

```
class Myexception extends Exception
{
    int detail;
    Myexception(int a)
    {
        detail = a;
    }
    public String toString()
    {
      return "detail";
    }
}
class Output
{
    static void compute (int a) throws Myexception
    {
      throw new Myexception(a);
    }
    public static void main(String args[])
    {
      try
      {
        compute(3);
      }
      catch(Exception e)
      {
        System.out.print("Exception");
      }
   }
}
```

a) 3
b) Exception
c) Runtime Error
d) Compilation Error

**Answer : B**

Explanation: Myexception is self defined exception.

**Question 484:** What is the output of this program?

```
class exception_handling
{
   public static void main(String args[])
   {
      try
      {
         int a = args.length;
         int b = 10 / a;
         System.out.print(a);
         try
         {
            if (a == 1)
               a = a / a - a;
            if (a == 2)
            {
               int c = {1};
               c[8] = 9;
            }
         }
         catch (ArrayIndexOutOfBoundsException e)
         {
            System.out.println("TypeA");
         }
         catch (ArithmeticException e)
         {
            System.out.println("TypeB");
         }
      }
   }
}
```

Note : Execution command line : $ java exception_handling one

a) TypeA
b) TypeB
c) Compilation Error
d) Runtime Error

**Answer : C**

Explanation: try without catch or finally

# isAlive(), Join() & Thread Synchronization

**Question 485:** Which of this method can be used to make the main thread to be executed last among all the threads?

a) stop()
b) sleep()
c) join()
d) call()

**Answer : B**

Explanation: By calling sleep() within main(), with long enough delay to ensure that all child threads terminate prior to the main thread.

**Question 486:** Which of this method is used to find out that a thread is still running or not?

a) run()
b) Alive()
c) isAlive()
d) checkRun()

**Answer : C**

Explanation: The isAlive( ) method returns true if the thread upon which it is called is still running. It returns false otherwise.

**Question 487:** What is the default value of priority variable MIN_PRIORITY AND MAX_PRIORITY?

a) 0 & 256
b) 0 & 1
c) 1 & 10
d) 1 & 256

**Answer : C**

Explanation: None.

**Question 488:** Which of these method waits for the thread to terminate?

a) sleep()
b) isAlive()
c) join()
d) stop()

**Answer : C**

Explanation: None.

**Question 489:** Which method is used to explicitly set the priority of a thread?

a) set()
b) make()
c) setPriority()
d) makePriority()

**Answer : C**

Explanation: The default value of priority given to a thread is 5 but we can explicitly change that value between the permitted values 1 & 10, this is done by using the method setPriority().

**Question 490:** What is synchronization in reference to a thread?

a) It's a process of handling situations when two or more threads need access to a shared resource
b) It's a process by which many thread are able to access same shared resource simultaneously
c) It's a process by which a method is able to access many different threads simultaneously
d) It's a method that allow too many threads to access any information require

**Answer : A**

Explanation: When two or more threads need to access the same shared resource, they need some way to ensure that the resource will be used by only one thread at a time, the process by which this is achieved is called synchronization.

**Question 491:** What is the output of this program?

```
class newthread extends Thread
{
    newthread()
   {
        super("My Thread");
        start()
    }
    public void run()
    {
        System.out.println(this);
    }
}
class multithreaded_programing
{
   public static void main(String args[])
   {
      new newthread();
   }
}
```

a) My Thread
b) Thread[My Thread,5,main].
c) Compilation Error
d) Runtime Error

**Answer : B**

Explanation: Although we have not created any object of thread class still we can make a thread pointing to main method, we can refer it by using this.

**Question 492:** What is the output of this program?

```
class newthread extends Thread
{
    Thread t;
    newthread()
    {
        t = new Thread(this,"My Thread");
        t.start();
    }
    public void run()
    {
     try
     {
          t.join()
          System.out.println(t.getName());
     }
     catch(Exception e)
     {
          System.out.println("Exception");
     }
     }
}
class multithreaded_programing
{
   public static void main(String args[])
   {
      new newthread();
   }
}
```

a) My Thread
b) Thread[My Thread,5,main].
c) Exception
d) Runtime Error

**Answer : D**

Explanation: join() method of Thread class waits for thread being called to finish or terminate, but here we have no condition which can terminate the thread, hence code 't.join()' leads to runtime error and nothing will be printed on the screen.

**Question 493:** What is the output of this program?

```
class newthread extends Thread
{
    Thread t;
    newthread()
    {
       t = new Thread(this,"New Thread");
       t.start();
    }
   public void run()
   {
      System.out.println(t.isAlive());
    }
}
class multithreaded_programing
{
   public static void main(String args[])
   {
      new newthread();
   }
}
```

a) 0
b) 1
c) true
d) false

**Answer : C**

Explanation: isAlive() method is used to check whether the thread being called is running or not, here thread is the main() method which is running till the program is terminated hence it returns true.



**Question 494:** What is the output of this program?

```
class newthread extends Thread
{
    Thread t1,t2;
    newthread()
    {
       t1 = new Thread(this,"Thread_1");
       t2 = new Thread(this,"Thread_2");
       t1.start();
       t2.start();
    }
```

```
        public void run()
        {
            t2.setPriority(Thread.MAX_PRIORITY);
            System.out.print(t1.equals(t2));
        }
    }
    class multithreaded_programing
    {
        public static void main(String args[])
        {
            new newthread();
        }
    }
```

a) true
b) false
c) truetrue
d) falsefalse

**Answer : D**

Explanation: This program was previously done by using Runnable interface, here we have used Thread class. This shows both the method are equivalent, we can use any of them to create a thread.

**Question 495:** Which method is used to implement Runnable interface?

a) stop()
b) run()
c) runThread()
d) stopThread()

**Answer : B**

Explanation: To implement Runnable interface, a class needs only to implement a single method called run().

**Question 496:** Which method is used to begin the execution of a thread?

a) run()
b) start()
c) runThread()
d) startThread()

**Answer : B**

Explanation: None.

**Question 497:** Which statement is incorrect?

a) A thread can be formed by implementing Runnable interface
b) A thread can be formed by a class that extends Thread class
c) start() method is used to begin execution of the thread
d) run() method is used to begin execution of a thread before start() method in special cases

**Answer : D**

Explanation: run() method is used to define the code that constitutes the new thread, it contains the code to be executed. start() method is used to begin execution of the thread that is execution of run(). run() itself is never used for starting execution of the thread.

**Question 498:** What is the output of this program?

```
class newthread implements Runnable
{
    Thread t;
    newthread()
    {
       t = new Thread(this,"My Thread");
       t.start();
    }
   public void run()
   {
       System.out.println(t.getName());
    }
}
class multithreaded_programing
{
   public static void main(String args[])
   {
      new newthread();
   }
}
```

a) My Thread
b) Thread[My Thread,5,main].
c) Compilation Error
d) Runtime Error

**Answer : A**

Explanation: None.

**Question 499:** What is the output of this program?

```
class newthread implements Runnable
{
     Thread t;
     newthread()
    {
        t = new Thread(this,"My Thread");
        t.start();
     }
     public void run()
    {
        System.out.println(t);
     }
}
class multithreaded_programing
{
   public static void main(String args[])
   {
      new newthread();
   }
}
```

a) My Thread
b) Thread[My Thread,5,main].
c) Compilation Error
d) Runtime Error

**Answer : B**

Explanation: None.

**Question 500:** What is the output of this program?

```
class newthread implements Runnable
{
    Thread t;
    newthread()
   {
       t = new Thread(this,"My Thread");
       t.start();
    }
}
class multithreaded_programing
{
   public static void main(String args[])
   {
      new newthread();
   }
}
```

a) My Thread
b) Thread[My Thread,5,main].
c) Compilation Error
d) Runtime Error

**Answer : C**

Explanation: Thread t has been made by using Runnable interface, hence it is necessary to use inherited abstract method run() method to specify instructions to be implemented on the thread, since no run() method is used it gives a compilation error.

**Question 501:** What is the output of this program?

```
class newthread implements Runnable
{
    Thread t;
    newthread()
    {
       t = new Thread(this,"New Thread");
       t.start();
    }
    public void run()
    {
       t.setPriority(Thread.MAX_PRIORITY);
       System.out.println(t);
    }
}
class multithreaded_programing
{
   public static void main(String args[])
   {
      new newthread();
   }
}
```



a) Thread[New Thread,0,main]
b) Thread[New Thread,1,main]
c) Thread[New Thread,5,main]
d) Thread[New Thread,10,main]

**Answer : D**

Explanation: Thread t has been made with default priority value 5 but in run method the priority has been explicitly changed to MAX_PRIORITY of class thread, that is 10 by code 't.setPriority(Thread.MAX_PRIORITY);' using the setPriority function of thread t.

**Question 502:** What is the output of this program?

```
class newthread implements Runnable
{
    Thread t;
    newthread()
    {
        t1 = new Thread(this,"Thread_1");
        t2 = new Thread(this,"Thread_2");
        t1.start();
        t2.start();
    }
    public void run()
    {
        t2.setPriority(Thread.MAX_PRIORITY);
        System.out.print(t1.equals(t2));
    }
}
class multithreaded_programing
{
    public static void main(String args[])
    {
        new newthread();
    }
}
```

a) true
b) false
c) truetrue
d) falsefalse

**Answer : D**

Explanation: Threads t1 & t2 are created by class newthread that is implementing runnable interface, hence both the threads are provided their own run() method specifying the actions to be taken. When constructor of newthread class is called first the run() method of t1 executes than the run method of t2 printing 2 times "false" as both the threads are not equal one is having different priority than other, hence falsefalse is printed.

**Question 503:** Which method of Thread class is used to find out the priority given to a thread?

a) get()
b) ThreadPriority()
c) getPriority()
d) getThreadPriority()

**Answer : C**

Explanation: None.

**Question 504:** Which method of Thread class is used to suspend a thread for a period of time?

a) sleep()
b) terminate()
c) suspend()
d) stop()

**Answer : A**

Explanation: None.

**Question 505:** Which function of pre defined class Thread is used to check weather current thread being checked is still running?

a) isAlive()
b) Join()
c) isRunning()
d) Alive()

**Answer : A**

Explanation:isAlive() function is defined in class Thread, it is used for implementing multithreading and to check whether the thread called upon is still running or not.

**Question 506:** What is the output of this program?

```
class multithreaded_programing
{
    public static void main(String args[])
    {
        Thread t = Thread.currentThread();
        t.setName("New Thread");
        System.out.println(t);
    }
}
```

a) Thread[5,main]
b) Thread[New Thread,5]
c) Thread[main,5,main]
d) Thread[New Thread,5,main]

**Answer : D**

Explanation: None.

**Question 507:** What is the priority of the thread in output of this program?

```
class multithreaded_programing
{
   public static void main(String args[])
   {
      Thread t = Thread.currentThread();
      t.setName("New Thread");
      System.out.println(t.getName());
   }
}
```

a) main
b) Thread
c) New Thread
d) Thread[New Thread,5,main]

**Answer : C**

Explanation: The getName() function is used to obtain the name of the thread, in this code the name given to thread is 'New Thread'.

**Question 508:** What is the name of the thread in output of this program?

```
class multithreaded_programing
{
   public static void main(String args[])
   {
      Thread t = Thread.currentThread();
      System.out.println(t.getPriority());
   }
}
```

a) 0
b) 1
c) 4
d) 5

**Answer : D**

Explanation: The default priority given to a thread is 5.

**Question 509:** What is the name of the thread in output of this program?

```
class multithreaded_programing
{
   public static void main(String args[])
   {
      Thread t = Thread.currentThread();
      System.out.println(t.isAlive());
   }
}
```

a) 0
b) 1
c) true
d) false

**Answer : C**

Explanation: Thread t is seeded to currently program, hence when you run the program the thread becomes active & code 't.isAlive' returns true.

**Question 510:** What is multithreaded programming?

a) It's a process in which two different processes run simultaneously
b) It's a process in which two or more parts of same process run simultaneously
c) It's a process in which many different process are able to access same information
d) It's a process in which a single process can access information from many sources

**Answer : B**

Explanation: Multithreaded programming a process in which two or more parts of the same process run simultaneously.

**Question 511:** Which of these are types of multitasking?

a) Process based
b) Thread based
c) Process and Thread based
d) None of the mentioned

**Answer : C**

Explanation: There are two types of multitasking: Process based multitasking and Thread based multitasking.

**Question 512:** Thread priority in Java is?

a) Integer
b) Float
c) double
d) long

**Answer : A**

Explanation: Java assigns to each thread a priority that determines hoe that thread should be treated with respect to others. Thread priority is integers that specify relative priority of one thread to another.

**Question 513:** What is the priority of the thread in output of this program?

```
class multithreaded_programing
{
   public static void main(String args[])
   {
      Thread t = Thread.currentThread();
      System.out.println(t);
   }
}
```

a) 4
b) 5
c) 0
d) 1

**Answer : B**

Explanation: The output of program is Thread[main,5,main], in this priority assigned to the thread is 5. It's the default value. Since we have not named the thread they are named by the group to they belong i.e main method.



**Question 514:** What requires less resources?

a) Thread
b) Process
c) Thread and Process
d) Neither Thread nor Process

**Answer : A**

Explanation: Thread is a lightweight and requires less resources to create and exist in the process. Thread shares the process resources.

**Question 515:** What does not prevent JVM from terminating?

a) Process
b) Daemon Thread
c) User Thread
d) JVM Thread

**Answer : B**

Explanation: Daemon thread runs in the background and does not prevent JVM from terminating. Child of daemon thread is also daemon thread.

**Question 516:** What decides thread priority?

a) Process
b) Process scheduler
c) Thread
d) Thread scheduler

**Answer : D**

Explanation: Thread scheduler decides the priority of the thread execution. This cannot guarantee that higher priority thread will be executed first, it depends on thread scheduler implementation that is OS dependent.

**Question 517:** What is true about time slicing?

a) Time slicing is OS service that allocates CPU time to available runnable thread
b) Time slicing is the process to divide the available CPU time to available runnable thread
c) Time slicing depends on its implementation in OS
d) Time slicing allocates more resources to thread

**Answer : B**

Explanation: Time slicing is the process to divide the available CPU time to available runnable thread.



**Question 518:** Deadlock is a situation when thread is waiting for other thread to release acquired object.

a) True
b) False

**Answer : A**

Explanation: Deadlock is java programming situation where one thread waits for an object lock that is acquired by other thread and vice-versa.

**Question 519:** What is true about threading?

a) run() method calls start() method and runs the code
b) run() method creates new thread
c) run() method can be called directly without start() method being called
d) start() method creates new thread and calls code written in run() method

**Answer : D**

Explanation: start() eventually calls run() method. Start() method creates thread and calls the code written inside run method.

**Question 520:** Which of the following is a correct constructor for thread?

a) Thread(Runnable a, String str)
b) Thread(int priority)
c) Thread(Runnable a, int priority)
d) Thread(Runnable a, ThreadGroup t)

**Answer : A**

Explanation: Thread(Runnable a, String str) is a valid constructor for thread. Thread() is also a valid constructor.

**Question 521:** Which of the following will ensure the thread will be in running state?

a) yield()
b) notify()
c) wait()
d) Thread.killThread()

**Answer : C**

Explanation: wait() always causes the current thread to go into the object's wait pool. Hence, using this in a thread will keep it in running state.

**Question 522:** Which of the following stops execution of a thread?

a) Calling stop() method on a Thread object
b) Calling notify() method on an object
c) Calling wait() method on an object
d) Calling read() method on an InputStream object

**Answer : B**

Explanation: notify() wakes up a single thread which is waiting for this object.

**Question 523:** Which keyword is used to implement synchronization?

a) synchronize
b) syn
c) synch
d) synchronized

**Answer : D**

Explanation: None.

**Question 524:** Which method is used to tell the calling thread to give up a monitor and go to sleep until some other thread enters the same monitor?

a) wait()
b) notify()
c) notifyAll()
d) sleep()

**Answer : A**

Explanation: wait() method is used to tell the calling thread to give up a monitor and go to sleep until some other thread enters the same monitor. This helps in avoiding polling and minimizes CPU idle time.

**Question 525:** Which method wakes up the first thread that called wait()?

a) wake()
b) notify()
c) start()
d) notifyAll()

**Answer : B**

Explanation: None.

**Question 526:** Which method wakes up all the threads?

a) wakeAll()
b) notify()
c) start()
d) notifyAll()

**Answer : D**

Explanation: notifyAll() wakes up all the threads that called wait() on the same object. The highest priority thread will run first.

**Question 527:** What is the output of this program?

```
class newthread extends Thread
{
    Thread t;
    String name;
    newthread(String threadname)
    {
       name = threadname;
       t = new Thread(this,name);
       t.start();
    }
    public void run()
    {  }
}
class multithreaded_programing
{
   public static void main(String args[])
   {
       newthread obj1 =  new newthread("one");
       newthread obj2 =  new newthread("two");
       try
       {
           obj1.t.wait();
           System.out.print(obj1.t.isAlive());
       }
       catch(Exception e)
       {
           System.out.print("Main thread interrupted");
       }
   }
}
```

a) true
b) false
c) Main thread interrupted
d) None of the mentioned

**Answer : C**

Explanation: obj1.t.wait() causes main thread to go out of processing in sleep state hence causes exception and “Main thread interrupted” is printed.

**Question 528:** What is the output of this program?

```
class newthread extends Thread
{
    Thread t;
    String name;
    newthread(String threadname)
    {
       name = threadname;
       t = new Thread(this,name);
       t.start();
    }
    public void run()
    {  }
}
class multithreaded_programing
{
   public static void main(String args[])
   {
       newthread obj1 =   new newthread("one");
       newthread obj2 =   new newthread("two");
       try
       {
           Thread.sleep(1000);
           System.out.print(obj1.t.isAlive());
       }
       catch(InterruptedException e)
       {
            System.out.print("Main thread interrupted");
       }
   }
}
```

a) true
b) false
c) Main thread interrupted
d) None of the mentioned

**Answer : B**

Explanation: Thread.sleep(1000) has caused all the threads to be suspended for some time, hence onj1.t.isAlive() returns false.

**Question 529:** What is the output of this program?

```
class newthread extends Thread
{
    Thread t;
    String name;
    newthread(String threadname)
    {
       name = threadname;
       t = new Thread(this,name);
       t.start();
    }
    public void run()
    {   }
}
class multithreaded_programing
{
   public static void main(String args[])
   {
        newthread obj1 =   new newthread("one");
        newthread obj2 =   new newthread("two");
        try
        {
            System.out.print(obj1.t.equals(obj2.t));
        }
       catch(Exception e)
       {
            System.out.print("Main thread interrupted");
       }
   }
}
```

a) true
b) false
c) Main thread interrupted
d) None of the mentioned

**Answer : B**

Explanation: Both obj1 and obj2 have threads with different name that is “one” and “two” hence obj1.t.equals(obj2.t) returns false.

**Question 530:** What is the output of this program?

```
class newthread extends Thread
{
    Thread t;
    newthread()
    {
        t1 = new Thread(this,"Thread_1");
        t2 = new Thread(this,"Thread_2");
        t1.start();
        t2.start();
    }
    public void run()
    {
        t2.setPriority(Thread.MAX_PRIORITY);
        System.out.print(t1.equals(t2));
    }
}
class multithreaded_programing
{
   public static void main(String args[])
   {
      new newthread();
   }
}
```

a) true
b) false
c) truetrue
d) falsefalse

**Answer : D**

Explanation: This program was previously done by using Runnable interface, here we have used Thread class. This shows both the method are equivalent, we can use any of them to create a thread.

# Input & Output Basics

**Question 531:** What does AWT stands for?

a) All Window Tools
b) All Writing Tools
c) Abstract Window Toolkit
d) Abstract Writing Toolkit

**Answer : C**

Explanation: AWT stands for Abstract Window Toolkit, it is used by applets to interact with the user.

**Question 532:** Which of these is used to perform all input & output operations in Java?

a) streams
b) Variables
c) classes
d) Methods

**Answer : A**

Explanation: Like in any other language, streams are used for input and output operations.

**Question 533:** What is the output of this program if input given is 'abcqfghqbcd'?

```
class Input_Output
{
   public static void main(String args[]) throws IOException
   {
      char c;
      BufferedReader obj = new BufferedReader(new InputStreamReader(System.in));
      do
      {
         c = (char) obj.read();
            System.out.print(c);
      } while(c != 'q');
   }
}
```

a) abcqfgh
b) abc
c) abcq
d) abcqfghq

**Answer : C**

Explanation: None.

# Reading & Writing Console

**Question 534:** Which exception is thrown by read() method?

a) IOException
b) InterruptedException
c) SystemException
d) SystemInputException

**Answer : A**

Explanation: read method throws IOException.

**Question 535:** Which of these methods can be used to writing console output?

a) print()
b) println()
c) write()
d) all of the mentioned

**Answer : D**

Explanation: None.

**Question 536:** Which of these classes are used by character streams output operations?

a) InputStream
b) Writer
c) ReadStream
d) InputOutputStream

**Answer : B**

Explanation: Character streams uses Writer and Reader classes for input & output operations.

**Question 537:** Which class is used to read from a file?

a) InputStream
b) BufferedInputStream
c) FileInputStream
d) BufferedFileInputStream

**Answer : C**

Explanation: None.

**Question 538:** What is the output of this program?

```
class output
{
   public static void main(String args[])
   {
      String a="hello i love java";
      System.out.println(indexof('i')+" "+indexof('o')+" "+lastIndexof('i')+" "+lastIndexof('o') ));
   }
}
```

a) 6 4 6 9
b) 5 4 5 9
c) 7 8 8 9
d) 4 3 6 9

**Answer : A**

Explanation: indexof('c') and lastIndexof('c') are pre defined function which are used to get the index of first and last occurrence of the character pointed by c in the given array.

**Question 539:** Which exception is thrown in cases when the file specified for writing is not found?

a) IOException
b) FileException
c) FileNotFoundException
d) FileInputException

**Answer : C**

Explanation: In cases when the file specified is not found, then FileNotFoundException is thrown by java run-time system, earlier versions of java used to throw IOException but after Java 2.0 they throw FileNotFoundException.

**Question 540:** Which of these methods are used to read in from file?

a) get()
b) read()
c) scan()
d) readFileInput()

**Answer : B**

Explanation: Each time read() is called, it reads a single byte from the file and returns the byte as an integer value. read() returns -1 when the end of the file is encountered.

**Question 541:** Which of these values is returned by read() method is end of file (EOF) is encountered?

a) 0
b) 1
c) -1
d) Null

**Answer : C**

Explanation: Each time read() is called, it reads a single byte from the file and returns the byte as an integer value. read() returns -1 when the end of the file is encountered.

**Question 542:** Which of these exception is thrown by close() and read() methods?

a) IOException
b) FileException
c) FileNotFoundException
d) FileInputOutputException

**Answer : A**

Explanation: Both close() and read() method throw IOException.

**Question 543:** Which of these methods is used to write() into a file?

a) put()
b) print()
c) write()
d) writeFile()

**Answer : C**

Explanation: None.

# Applets Fundamentals

**Question 544:** Which of these functions is called to display the output of an applet?

a) display()
b) paint()
c) displayApplet()
d) PrintApplet()

**Answer : B**

Explanation: Whenever the applet requires to redraw its output, it is done by using method paint().

**Question 545:** Which method can be used to output a string in an applet?

a) display()
b) print()
c) drawString()
d) transient()

**Answer : C**

Explanation: drawString() method is defined in Graphics class, it is used to output a string in an applet.

**Question 546:** Which method is a part of Abstract Window Toolkit (AWT) ?

a) display()
b) paint()
c) drawString()
d) transient()

**Answer : B**

Explanation: paint() is an abstract method defined in AWT.



**Question 547:** What is the Message is displayed in the applet made by this program?

```
import java.awt.*;
import java.applet.*;
public class myapplet extends Applet
{
   public void paint(Graphics g)
   {
      g.drawString("A Simple Applet", 20, 20);
   }
}
```

a) A Simple Applet
b) A Simple Applet 20 20
c) Compilation Error
d) Runtime Error

**Answer : A**

Explanation: None.

**Question 548:** What is the output of this program?

```
import java.awt.*;
import java.applet.*;
public class myapplet extends Applet
{
   Graphic g;
   g.drawString("20", 20, 20);
}
```

a) 20 20 20
b) "20" 20 20
c) Compilation Error
d) 20

**Answer : C**

Explanation: To implement the method drawString we need first need to define abstract method of AWT that is paint() method. Without paint() method we can not define and use drawString or any Graphic class methods.

**Question 549:** Which of these packages contains all the classes and methods required for even handling in Java?

a) java.applet
b) java.awt
c) java.event
d) java.awt.event

**Answer : D**

Explanation: Most of the event to which an applet response is generated by a user. Hence they are in Abstract Window Kit package, java.awt.event.

**Question 550:** What is an event in delegation event model used by Java programming language?

a) An event is an object that describes a state change in a source
b) An event is an object that describes a state change in processing
c) An event is an object that describes any change by the user and system
d) An event is a class used for defining object, to create events

**Answer : A**

Explanation: An event is an object that describes a state change in a source.

**Question 551:** Which method is used to register a keyboard event listener?

a) KeyListener()
b) addKistener()
c) addKeyListener()
d) eventKeyboardListener()

**Answer : C**

Explanation: None.

**Question 552:** Which method is used to register a mouse motion listener?

a) addMouse()
b) addMouseListener()
c) addMouseMotionListner()
d) eventMouseMotionListener()

**Answer : C**

Explanation: None.

**Question 553:** What is a listener in context to event handling?

a) A listener is a variable that is notified when an event occurs
b) A listener is a object that is notified when an event occurs
c) A listener is a method that is notified when an event occurs
d) None of the mentioned

**Answer : B**

Explanation: A listener is a object that is notified when an event occurs. It has two major requirements first, it must have been registered with one or more sources to receive notification about specific event types, and secondly it must implement methods to receive and process these notifications.



**Question 554:** Which method can be used to determine the type of event?

a) getID()
b) getSource()
c) getEvent()
d) getEventObject()

**Answer : A**

Explanation: getID() can be used to determine the type of an event.

**Question 555:** Which class is super class of all the events?

a) EventObject
b) EventClass
c) ActionEvent
d) ItemEvent

**Answer : A**

Explanation: EventObject class is a super class of all the events and is defined in java.util package.

**Question 556:** Which of these events will be notified if scroll bar is manipulated?

a) ActionEvent
b) ComponentEvent
c) AdjustmentEvent
d) WindowEvent

**Answer : C**

Explanation: AdjustmentEvent is generated when a scroll bar is manipulated.

**Question 557:** Which of these events will be generated if we close an applet's window?

a) ActionEvent
b) ComponentEvent
c) AdjustmentEvent
d) WindowEvent

**Answer : D**

Explanation: WindowEvent is generated when a window is activated, closed, deactivated, deiconfied, iconfied, opened or quit.

**Question 558:** Which of these events is generated when a button is pressed?

a) ActionEvent
b) KeyEvent
c) WindowEvent
d) AdjustmentEvent

**Answer : A**

Explanation: Action event is generated when a button is pressed, a list item is double-clicked or a menu item is selected.

**Question 559:** Which of these methods can be used to obtain the command name for invoking ActionEvent object?

a) getCommand()
b) getActionCommand()
c) getActionEvent()
d) getActionEventCommand()

**Answer : B**

Explanation: None.

**Question 560:** Which integer constants defined in ActionEvent class?

a) ALT_MASK
b) CTRL_MASK
c) SHIFT_MASK
d) All of the mentioned

**Answer : D**

Explanation: Action event defines 4 integer constants ALT_MASK, CTRL_MASK, SHIFT_MASK and ACTION_PERFORMED.

**Question 561:** Which of these methods can be used to know which key is pressed?

a) getKey()
b) getModifier()
c) getActionKey()
d) getActionEvent()

**Answer : B**

Explanation: The getModifiers() methods returns a value that indicates which modifiers keys (ALT, CTRL, META, SHIFT) were pressed when the event was generated.

**Question 562:** Which of these events is generated when the size of an event is changed?

a) ComponentEvent
b) ContainerEvent
c) FocusEvent
d) InputEvent

**Answer : A**

Explanation: A ComponentEvent is generated when the size, position or visibility of a component is changed.

**Question 563:** Which of these events is generated when the component is added or removed?

a) ComponentEvent
b) ContainerEvent
c) FocusEvent
d) InputEvent

**Answer : B**

Explanation: A ContainerEvent is generated when a component is added to or removed from a container. It has two integer constants COMPONENT_ADDED & COMPONENT_REMOVED.

**Question 564:** Which of these methods can be used to get reference to a component that was removed from a container?

a) getComponent()
b) getchild()
c) getContainerComponent()
d) getComponentChild()

**Answer : B**

Explanation: The getChild() method returns a reference to the component that was added to or removed from the container.

**Question 565:** Which of these events is generated when computer gains or loses input focus?

a) ComponentEvent
b) ContainerEvent
c) FocusEvent
d) InputEvent

**Answer : C**

Explanation: None.

**Question 566:** FocusEvent is subclass of which of these classes?

a) ComponentEvent
b) ContainerEvent
c) ItemEvent
d) InputEvent

**Answer : A**

Explanation: None.

**Question 567:** Which of these events is generated when the window is closed?

a) TextEvent
b) MouseEvent
c) FocusEvent
d) WindowEvent

**Answer : D**

Explanation: A WindowEvent is generated when a window is opened, close, activated or deactivated.

**Question 568:** Which of these methods can be used to obtain the coordinates of a mouse?

a) getPoint()
b) getCoordinates()
c) getMouseXY()
d) getMouseCordinates()

**Answer : A**

Explanation: getPoint() method can be used to obtain coordinates of a mouse, alternatively we can use getX() and getY() methods for x and y coordinates of mouse respectively.

**Question 569:** Which of these are integer constants of TextEvent class?

a) TEXT_CHANGED
b) TEXT_FORMAT_CHANGED
c) TEXT_VALUE_CHANGED
d) TEXT_sIZE_CHANGED

**Answer : C**

Explanation: TextEvent defines a single integer constant TEXT_VALUE_CHANGED.

**Question 570:** MouseEvent is subclass of which of these classes?

a) ComponentEvent
b) ContainerEvent
c) ItemEvent
d) InputEvent

**Answer : D**

Explanation: None.

**Question 571:** Which of these methods is used to get x coordinate of the mouse?

a) getX()
b) getXCoordinate()
c) getCoordinateX()
d) getPointX()

**Answer : A**

Explanation: getX() and getY() are used to obtain X AND Y coordinates of the mouse.

**Question 572:** Which of these are constants defined in WindowEvent class?

a) WINDOW_ACTIVATED
b) WINDOW_CLOSED
c) WINDOW_DEICONIFIED
d) All of the mentioned

**Answer : D**

Explanation: WindowEvent class defines 7 constants – WINDOW_ACTIVATED, WINDOW_CLOSED, WINDOW_OPENED, WINDOW_DECONIFIED, WINDOW_CLOSING, WINDOW_DEACTIVATED, WINDOW_ICONIFIED.

**Question 573:** Which of these is superclass of WindowEvent class?

a) WindowEvent
b) ComponentEvent
c) ItemEvent
d) InputEvent

**Answer : B**

Explanation: ComponentEvent is superclass of ContainerEvent, FocusEvent, KeyEvent, MouseEvent and WindowEvent.

**Question 574:** Which of these interfaces handles the event when a component is added to a container?

a) ComponentListener
b) ContainerListener
c) FocusListener
d) InputListener

**Answer : B**

Explanation: The ContainerListener defines methods to recognize when a component is added to or removed from a container.

**Question 575:** Which of these interfaces define a method actionPerformed()?

a) ComponentListener
b) ContainerListener
c) ActionListener
d) InputListener

**Answer : C**

Explanation: ActionListener defines the actionPerformed() method that is invoked when an adjustment event occurs.

**Question 576:** Which of these interfaces define a method itemStateChanged()?

a) ComponentListener
b) ContainerListener
c) ActionListener
d) ItemListener

**Answer : D**

Explanation: None.

**Question 577:** Which of these methods will be invoked if a character is entered?

a) keyPressed()
b) keyReleased()
c) keyTyped()
d) keyEntered()

**Answer : C**

Explanation: None.

**Question 578:** Which of these methods is defined in MouseMotionAdapter class?

a) mouseDragged()
b) mousePressed()
c) mouseReleased()
d) mouseClicked()

**Answer : A**

Explanation: The MouseMotionAdapter class defines 2 methods – mouseDragged() and mouseMoved().

**Question 579:** Which of these is a superclass of all Adapter classes?

a) Applet
b) ComponentEvent
c) Event
d) InputEvent

**Answer : A**

Explanation: All Adapter classes extend Applet class.

# Random Number

**Question 580:** Which class is used to generate random number?

a) java.lang.Object
b) java.util.randomNumber
c) java.util.Random
d) java.util.Object

**Answer : C**

Explanation: java.util.random class is used to generate random numbers in java program.

**Question 581:** Which method is used to generate boolean random values in java?

a) nextBoolean()
b) randomBoolean()
c) previousBoolean()
d) generateBoolean()

**Answer : A**

Explanation: nextBoolean() method of java.util.Random class is used to generate random numbers.

**Question 582:** What is the return type of Math.random() method?

a) Integer
b) Double
c) String
d) Boolean

**Answer : B**

Explanation: Math.random() method returns floating point number or precisely a double.

**Question 583:** What is the range of numbers returned by Math.random() method?

a) -1.0 to 1.0
b) -1 to 1
c) 0 to 100
d) 0.0 to 1.0

**Answer : D**

Explanation: Math.random() returns only double value greater than or equal to 0.0 and less than 1.0.

# Packages

**Question 584:** Which keyword is used to define packages in Java?

a) pkg
b) Pkg
c) package
d) Package

**Answer : C**

Explanation: None.

**Question 585:** Which of these is a mechanism for naming and visibility control of a class and its content?

a) Object
b) Packages
c) Interfaces
d) None of the Mentioned

**Answer : B**

Explanation: Packages are both naming and visibility control mechanism. We can define a class inside a package which is not accessible by code outside the package.

**Question 586:** Which of this access specifies can be used for a class so that its members can be accessed by a different class in the same package?

a) Public
b) Protected
c) No Modifier(Default)
d) All of the mentioned

**Answer : D**

Explanation: Either we can use public, protected or we can name the class without any specifier.

**Question 587:** Which of these access specifiers can be used for a class so that its members can be accessed by a different class in the different package?

a) Public
b) Protected
c) Private
d) No Modifier

**Answer : A**

Explanation: None.

**Question 588:** Which of the following is the correct way of importing an entire package 'pkg'?

a) import pkg;
b) Import pkg.
c) import pkg.*
d) Import pkg.*

**Answer : C**

Explanation: Operator * is used to import the entire package.

**Question 589:** Which of the following package stores all the standard java classes?

a) lang
b) java
c) util
d) java.packages

**Answer : B**

Explanation: None.

**Question 590:** What is the output of this program?

```
package pkg;
class display
{
   int x;
   void show()
   {
      if (x > 1)
         System.out.print(x + " ");
   }
}
class packages
{
   public static void main(String args[])
   {
      display[] arr=new display[3];
      for(int i=0;i<3;i++)
         arr[i]=new display();
      arr[0].x = 0;
      arr[1].x = 1;
      arr[2].x = 2;
      for (int i = 0; i < 3; ++i)
         arr[i].show();
    }
}
```

Note : packages.class file is in directory pkg;

a) 0
b) 1
c) 2
d) 0 1 2

**Answer : C**

Explanation: None.

**Question 591:** What is the output of this program?

```
package pkg;
class output
{
   public static void main(String args[])
   {
      StringBuffer s1 = new StringBuffer("Hello");
      s1.setCharAt(1, x);
      System.out.println(s1);
   }
}
```

a) xello
b) xxxxx
c) Hxllo
d) Hexlo

**Answer : C**

Explanation: None.

# Core Java API Packages

**Question 592:** Which package is used for graphical user interface?

a) java.applet
b) java.awt
c) java.awt.image
d) java.io

**Answer : B**

Explanation: java.awt provides capabilities for graphical user interface.

**Question 593:** Which of this package is used for analyzing code during run-time?

a) java.applet
b) java.awt
c) java.io
d) java.lang.reflect

**Answer : D**

Explanation: Reflection is the ability of software to analyze itself. This is provided by java.lang.reflect package.

**Question 594:** Which of this package is used for handling security related issues in a program?

a) java.security
b) java.lang.security
c) java.awt.image
d) java.io.security

**Answer : A**

Explanation: java.security handles certificates, keys, digests, signatures, and other security functions.

# Interfaces

**Question 595:** Which keyword is used to define interfaces in Java?

a) interface
b) Interface
c) intf
d) Intf

**Answer : A**

Explanation: None.

**Question 596:** Which of these can be used to fully abstract a class from its implementation?

a) Objects
b) Packages
c) Interfaces
d) Final

**Answer : C**

Explanation: None.

**Question 597:** Which of these access specifiers can be used for an interface?

a) Public
b) Protected
c) private
d) All of the mentioned

**Answer : A**

Explanation: Access specifier of an interface is either public or no specifier. When no access specifier is used then default access specifier is used due to which interface is available only to other members of the package in which it is declared, when declared public it can be used by any code.



**Question 598:** Which keyword is used by a class to use an interface defined previously?

a) import
b) Import
c) implements
d) Implements

**Answer : C**

Explanation: interface is inherited by a class using implements.

**Question 599:** Which of the following is the correct way of implementing an interface salary by class manager?

a) class manager extends salary {}
b) class manager implements salary {}
c) class manager imports salary {}
d) none of the mentioned

**Answer : B**

Explanation: None.

**Question 600:** Which of the following is an incorrect statement about interfaces?

a) Interfaces specifies what class must do but not how it does
b) Interfaces are specified public if they are to be accessed by any code in the program
c) All variables in interface are implicitly final and static.
d) All variables are static and methods are public if interface is defined pubic

**Answer : D**

Explanation: All methods and variables are implicitly public if interface is declared public.

**Question 601:** What is the output of this program?

```
interface calculate
{
   void cal(int item);
}
class display implements calculate
{
   int x;
   public void cal(int item)
   {
      x = item * item;
   }
}
class interfaces
{
   public static void main(String args[])
   {
      display arr = new display;
      arr.x = 0;
      arr.cal(2);
      System.out.print(arr.x);
   }
}
```

a) 0
b) 2
c) 4
d) None of the mentioned

**Answer : C**

Explanation: None.

**Question 602:** What is the output of this program?

```
interface calculate
{
   void cal(int item);
}
class displayA implements calculate
{
   int x;
   public void cal(int item)
   {
      x = item * item;
   }
}
class displayB implements calculate
{
   int x;
   public void cal(int item)
   {
      x = item / item;
   }
}
class interfaces
{
   public static void main(String args[])
   {
      displayA arr1 = new displayA;
      displayB arr2 = new displayB;
      arr1.x = 0;
      arr2.x = 0;
      arr1.cal(2);
      arr2.cal(2);
      System.out.print(arr1.x + " " + arr2.x);
   }
}
```

a) 0 0
b) 2 2
c) 4 1
d) 1 4

**Answer : C**

Explanation: class displayA implements the interface calculate by doubling the value of item, where as class displayB implements the interface by dividing item by item, therefore variable x of class displayA stores 4 and variable x of class displayB stores 1.

**Question 603:** What is the output of this program?

```
interface calculate
{
        int VAR = 0;
        void cal(int item);
}
class display implements calculate
{
        int x;
       public  void cal(int item)
       {
           if (item<2)
              x = VAR;
           else
              x = item * item;
        }
}
class interfaces
{
        public static void main(String args[])
        {
          display[] arr=new display[3];
          for(int i=0;i<3;i++)
                    arr[i]=new display();
          arr[0].cal(0);
          arr[1].cal(1);
          arr[2].cal(2);
          System.out.print(arr[0].x+" " + arr[1].x + " " + arr[2].x);
        }
}
```

a) 0 1 2
b) 0 2 4
c) 0 0 4
d) 0 1 4

**Answer : C**

Explanation: None.

**Question 604:** What does an interface contain?

a) Method definition
b) Method declaration
c) Method declaration and definition
d) Method name

**Answer : B**

Explanation: Interface contains the only declaration of the method.

**Question 605:** What type of methods an interface contain by default?

a) abstract
b) static
c) final
d) private

**Answer : A**

Explanation: By default, interface contains abstract methods. The abstract methods need to be implemented by concrete classes.

**Question 606:** What will happen if we provide concrete implementation of method in interface?

a) The concrete class implementing that method need not provide implementation of that method
b) Runtime exception is thrown
c) Compilation failure
d) Method not found exception is thrown

**Answer : C**

Explanation: The methods of interfaces are always abstract. They provide only method declaration.

**Question 607:** What happens when a constructor is defined for an interface?

a) Compilation failure
b) Runtime Exception
c) The interface compiles successfully
d) The implementing class will throw exception

**Answer : A**

Explanation: Constructor is not provided by interface as objects cannot be instantiated.

**Question 608:** What happens when we access the same variable defined in two interfaces implemented by the same class?

a) Compilation failure
b) Runtime Exception
c) The JVM is not able to identify the correct variable
d) The interfaceName.variableName needs to be defined

**Answer : D**

Explanation: The JVM needs to distinctly know which value of variable it needs to use. To avoid confusion to the JVM interfaceName.variableName is mandatory.

# Generics

**Question 609:** What is the output of this program?

```
import java.util.*;
public class genericstack <E>
{
    Stack <E> stk = new Stack <E>();
    public void push(E obj)
    {
        stk.push(obj);
    }
    public E pop()
    {
        E obj = stk.pop();
        return obj;
    }
}
class Output
{
   public static void main(String args[])
   {
      genericstack <String> gs = new genericstack<String>();
      gs.push("Hello");
      System.out.println(gs.pop());
   }
}
```

a) H
b) Hello
c) Runtime Error
d) Compilation Error

**Answer : B**

Explanation: None.

**Question 610:** What is the output of this program?

```
import java.util.*;
public class genericstack <E>
{
    Stack <E> stk = new Stack <E>();
    public void push(E obj)
    {
        stk.push(obj);
    }
```

```
        public E pop()
        {
            E obj = stk.pop();
            return obj;
        }
    }
    class Output
    {
       public static void main(String args[])
       {
          genericstack <String> gs = new genericstack<String>();
          gs.push("Hello");
          System.out.print(gs.pop() + " ");
          genericstack <Integer> gs = new genericstack<Integer>();
          gs.push(36);
          System.out.println(gs.pop());
       }
    }
```

a) Error
b) Hello
c) 36
d) Hello 36

**Answer : D**

Explanation: None.

**Question 611:** What is the output of this program?

```
import java.util.*;
public class genericstack <E>
{
    Stack <E> stk = new Stack <E>();
    public void push(E obj)
    {
        stk.push(obj);
    }
    public E pop()
    {
        E obj = stk.pop();
        return obj;
    }
}
```

```
class Output
{
   public static void main(String args[])
   {
      genericstack <Integer> gs = new genericstack<Integer>();
      gs.push("Hello");
      System.out.println(gs.pop());
   }
}
```

a) H
b) Hello
c) Runtime Error
d) Compilation Error

**Answer : D**

Explanation: genericstack's object gs is defined to contain a string parameter but we are sending an integer parameter, which results in compilation error.

**Question 612:** Which of the following is Type parameterized?

a) Overloaded Methods
b) Generic methods
c) Class methods
d) Overriding methods

**Answer : B**

Explanation: None.

**Question 613:** What are generic methods?

a) Generic methods are the methods defined in a generic class
b) Generic methods are the methods that extend generic class methods
c) Generic methods are methods that introduce their own type parameters
d) Generic methods are methods that take void parameters

**Answer : C**

Explanation: Generic methods are methods that introduce their own type parameters. This is similar to declaring a generic type, but the type parameter scope is limited to the method where it is declared. Static and non-static generic methods are allowed, as well as generic class constructors.

**Question 614:** What is the output of this program?

```
import java.util.*;
class Output
{
   public static double sumOfList(List<? extends Number> list)
   {
      double s = 0.0;
      for (Number n : list)
            s += n.doubleValue();
      return s;
   }
   public static void main(String args[])
   {
     List<Double> ld = Arrays.asList(1.2, 2.3, 3.5);
     System.out.println(sumOfList(ld));
   }
}
```

a) 5.0
b) 7.0
c) 8.0
d) 6.0

**Answer : B**

Explanation: None.

**Question 615:** Which of these is wildcard symbol?

a) ?
b) !
c) %
d) &

**Answer : A**

Explanation: In generic code, the question mark (?), called the wildcard, represents an unknown type.

**Question 616:** What is use of wildcards?

a) It is used in cases when type being operated upon is not known
b) It is used to make code more readable
c) It is used to access members of super class
d) It is used for type argument of generic method

**Answer : A**

Explanation: The wildcard can be used in a variety of situations: as the type of a parameter, field, or local variable; sometimes as a return type (though it is better programming practice to be more specific). The wildcard is never used as a type argument for a generic method invocation, a generic class instance creation, or a supertype.

**Question 617:** Which of these keywords is used to upper bound a wildcard?

a) stop
b) bound
c) extends
d) implements

**Answer : C**

Explanation: None.

**Question 618:** Which of these is an correct way making a list that is upper bounded by class Number?

a) List<? extends Number>
b) List<extends ? Number>
c) List(? extends Number)
d) List(? UpperBounds Number)

**Answer : A**

Explanation: None.

**Question 619:** Which of the following keywords are used for lower bounding a wild card?

a) extends
b) super
c) class
d) lower

**Answer : B**

Explanation: A lower bounded wildcard is expressed using the wildcard character ('?'), following by the super keyword, followed by its lower bound.