core java

****

Q. Memory deallocation in java is done by?
1). Programmer
2). Operating system
3). Garbage collector
4). None of the above
Solution :
option [3] is correct

Q. What is true about the following code?

1. enum EnumDemo { A }
2. class Test {
3. enum EnumD { B }
4. void my_method() {
5. enum EnumC { D }
6. } }
1). The code compiles without any error
2). The code compiles if we remove line number 1
3). The code compiles if we remove line number 5
4). None of the above is correct
Solution :
option [3] is correct

Q. Which of the following statements are true?
1). No arg constructor is allways supplied by the Compiler
2). Constructors cannot be overloaded
3). Constructors cannot have return type.
4). Constructors can be static
Solution :
option [3] is correct

Q. Which of the following are features of Java Programming Language?
1). Robust
2). Multithreaded
3). Interpreted
4). All of the above
Solution :
option [4] is correct

Q. public class test {

```
  static void methodA(short s) {
   System.out.println("methodA(short) called");
  }
```

```
   static void methodA(int i) {
    System.out.println("methodA(int) called");
   }


   static void methodB(float f) {
   System.out.println("methodB(float) called");
   }

   static void methodB(double d) {
   System.out.println("methodB(double) called");
   }

   public static void main(String args[]) {

   methodA(5);
   methodB(5.2);

   }
}
```

1). methodA(short) called
2). methodA(int) called
3). Exception
4). Compilation Error

Q. What all gets printed when the following code is compiled and run?
Select the correct answers:

```
public class Test {
  public static void main(String args[])
{
    for(int i = 0; i < 2; i++)
{
      for(int j = 2; j>= 0; j--)
{
        if(i == j) break;
       System.out.println("i=" + i + " j="+j);
      }
    }
  }
}
```

1. i=0 j=0
2. i=0 j=1
3. i=0 j=2
4. i=1 j=0
5. i=1 j=1

6. i=1 j=2
1). 1,3,5
2). 2,4,6
3). 1,2,5
4). 3,2,6
Solution :
option [4] is correct

Q. When access modifier is omitted from the definition of the member of a class. The member has?
1). default access
2). public access
3). private access
4). protected access
Solution :
option [1] is correct

Q. Given the following code fragment:

```
XXXX  choice ; // variable choice is declared and initialized here
switch( choice ) {
  case 100 : System.out.println("One hundred");break ;
  case 20 : System.out.println("Twenty");break ;
  case 30 : System.out.println("Thirty");break ;
}
```

Choose the declarations of choice which will not cause a compiler error.
1). byte choice = 100 ;
2). short choice = 100 ;
3). int choice = 300 ;
4). All of the above
Solution :
option [4] is correct

Q. Given the following code fragment:

```
XXXX  choice ; // variable choice is declared and initialized here
switch( choice ) {
  case 100 : System.out.println("One hundred");break ;
  case 20 : System.out.println("Twenty");break ;
  case 30 : System.out.println("Thirty");break ;
}
```

Choose the declarations of choice which will not cause a compiler error.
1). byte choice = 100 ;
2). short choice = 100 ;
3). int choice = 300 ;

4). All of the above
Solution :
option [4] is correct

Q. Given:

```
1. public class Demo {
2. public static void main (String[] args) {
3. byte var1 = 127;
4. byte var2 = 126;
5. byte result = var1 + var2;
6. }
7. }
```

Which statement is true?
1). Compilation succeeds and d takes the value 253.
2). Line 5 contains an error that prevents compilation.
3). Line 5 throws an exception indicating "Out of range"
4). Line 3 and 4 contain errors that prevent compilation.
Solution :
option [2] is correct

Q. Given:

```
  float result;
  result=5/2;
  System.out.println(result);
```

What will be the output?
1). 2
2). 2.0
3). 2.5
4). error
Solution :
option [2] is correct

Q. Given the following code fragment:

```
 XXXX  choice ; // variable choice is declared and initialized here
 switch( choice ) {
   case 100 : System.out.println("One hundred");break ;
   case 20 : System.out.println("Twenty");break ;
   case 30 : System.out.println("Thirty");break ;
 }
```

Choose the declarations of choice which will not cause a compiler error.
1). byte choice = 100 ;

2). short choice = 100 ;
3). int choice = 300 ;
4). All of the above
Solution :
option [4] is correct

Q. Integer var1 = new Integer(2);
Integer var2 = new Integer(2);

What happens when you do if (var1==var2)?
1). **'TRUE**
2). **'FALSE**
3). compilation error
4). runtime Exception
Solution :
option [2] is correct

Q. What will be the output of the following code snippet? '

```
String str1 = new String( "hello" );
String str2 = "hello";
        if  (str1==str2)
          System.out.println( "Equals");
        else
          System.out.println( "Not Equal");
```

1). Equal
2). **Not Equal**
3). **Compilation error**
4). None of the above
Solution :
option [2] is correct

Q. class Demo
{}
public class SubDemo extends Demo
{
public static void main(String []args){
Demo obj = new SubDemo();
System.out.println(obj instanceof Demo);}
}What will be the output of above code?
1). FALSE
2). **TRUE**
3). Demo
4). **SubDemo**

Solution :
option [2] is correct

Q. Which of the given syntax is correct to display date-time with the time zone in the ISO-8601 calendar system, such as 2007-12-03T10:15:30+01:00 Europe/Paris.
Assume the zone is Asia/kolkata.

1). **ZonedDateTime objt = ZonedDateTime.of(LocalDateTime.now(), ZoneId.of("Asia/Kolkata"));**

2). **LocalDate date = LocalDate.now(ZoneId.of("Asia/Kolkata"));**

3). ZonedDateTime time = new ZonedDateTime("Asia/Kolkata");

4). ZonedDateTime time = new ZonedDateTime.of("Asia/Kolkata");

Solution :
option [1] is correct

Q. The Scanner class is found in _____ package.

1). java.lang

2). **java.util**

3). java.io

4). None of the above

Solution :
option [2] is correct

Q. class Shape
{
final public double calArea(){}
}
public class Circle extends Shape
{
int radius;
public Circle(int radius){
this.radius = radius;
}
public double calArea(){
return 3.142*radius*radius;
}
public static void main(String []args){
Shape obj = new Circle(5);
System.out.println(obj.calArea());
}
}What will be the output of above code?

1). It will display area of circle with radius 5

2). no output

3). **compilation error**

4). runtime Exception

Solution :
option [3] is correct

Q. What will happen when you attempt to compile and run the following class?

```
class Base{
Base(int var)
{
   System.out.println("Base");
}
}
class Test extends Base{
public static void main(String argv[]){
       Test obj = new Test();
}
}
```

1). **Compiles without any problem**

2). Compiles and produces output "Base"

3). **Generates Compile time error**

4). None of the above

Solution :
option [3] is correct

Q. Which of the given LocalDate class method is used to obtain current date?

1). **now()**

2). parse()

3). **format()**

4). newInstance()

Solution :
option [1] is correct

Q.
```
class Employee
{
String name;
int id;
public Employee(String name,int id)
{
this.name = name;
this.id=id;
}
}
public class Manager extends Employee
{
```

```
public static void main(String []args)
{
Manager mgr = new Manager();
}
}
```
What will happen after execution of above code?

1). Manager class object will be successfully created

2). **compile time error as there is no default constructor in class Employee**

3). Manager object will be successfully created after implementing default constructor in Manager cla ss

4). Default constructor need to be implemented in both classes for successful creation of Manager cl ass object

Solution :
option [2] is correct

Q.
```
class Shape
{
final public double calArea(){}
}
public class Circle extends Shape
{
int radius;
public Circle(int radius){
this.radius = radius;
}
public double calArea(){
return 3.142*radius*radius;
}
public static void main(String []args){
Shape obj = new Circle(5);
System.out.println(obj.calArea());
}
}
```
What will be the output of above code?

1). It will display area of circle with radius 5

2). no output

3). **compilation error**

4). runtime Exception

Solution :
option [3] is correct

Q. What will happen when you attempt to compile and run the following class?

```
class Base{
Base(int var)
{
   System.out.println("Base");
}
```

```
}
class Test extends Base{
public static void main(String argv[]){
        Test obj = new Test();
}
}
```

1). **Compiles without any problem**

2). Compiles and produces output "Base"

3). **Generates Compile time error**

4). None of the above

Solution :
option [3] is correct

Q. Which of the given LocalDate class method is used to obtain current date?

1). **now()**

2). parse()

3). **format()**

4). newInstance()

Solution :
option [1] is correct

Q.
```
class Employee
{
String name;
int id;
public Employee(String name,int id)
{
this.name = name;
this.id=id;
}
}
public class Manager extends Employee
{
public static void main(String []args)
{
Manager mgr = new Manager();
}
}
```
What will happen after execution of above code?

1). Manager class object will be successfully created

2). **compile time error as there is no default constructor in class Employee**

3). Manager object will be successfully created after implementing default constructor in Manager cla ss

4). Default constructor need to be implemented in both classes for successful creation of Manager cl ass object

Solution :
option [2] is correct

Q. Which of the following are Object class methods?
1). **toString()**

2). equals()

3). hashCode()

4). **All of the above**

Solution :
option [4] is correct

Q. Which of the given syntax is correct for interface implementation in Java 8?
1). **interface Demo{**
}

2). **interface Demo{**
default int getNumber(){return 0;}
}

3). **interface Demo{**
static int getNumber(){return 0;}
}

4). interface Demo
{
int getNumber(){return 0;}
}

Solution :
option [1,2,3] are correct

Q. Which of the following is FALSE about abstract classes in Java?
1). If we derive an abstract class and do not implement all the abstract methods, then the derived cla ss should also be marked as abstract using 'abstract' keyword

2). Abstract classes can have constructors

3). **A class can be made abstract without any abstract method**

4). **A class can inherit from multiple abstract classes**

Solution :
option [4] is correct

Q. Consider the following code:

```
interface Greek { }

class Alpha implements Greek { }

class Beta extends Alpha {}

class Delta extends Beta
 {
   public static void main( String[] args )
    {
      Beta obj = new Beta(); // insert code here
    }
 }
```

Which of the following code snippet when inserted individual at the commented line (// insert code here), will cause a java.lang.ClassCastException?

1). Greek objGrk = (Beta)(Alpha)obj;

2). Alpha objAlpha = obj;

3). Greek objGrk = (Alpha)obj;

4). Beta objBeta = (Beta)(Alpha)obj;

5). **Greek objGrk = (Delta)obj;**

Solution :
option [5] is correct

Q.

```
public class Demos {

      public static void main(String[] args) {
       String input = "Hello Welcome";
       String pattern = "\\sHello\\sWelcome\\s";
       boolean flag = Pattern.matches(pattern, input);
       System.out.println(flag);
      }
}
```

1). **TRUE**

2). **FALSE**

3). compilation error

4). exception

Solution :
option [2] is correct

Q. class Shape
{
Shape(){}
}
public class Triangle extends Shape
{
int base,height;
public Triangle(int base,int height){
this.base = base;
this.height=height;
}
public double calArea(){
return o.5*base*height;
}
public static void main(String []args){
Shape ref = new Triangle(3,4);
System.out.println(ref.calArea());
}Which change need to be done in the Shape class for successful execution of program?

1). Declare Shape as abstract class

2). Remove default constructor from Shape class

3). declare abstract double calArea() in Shape class

4). **declare abstract double calArea() in Shape class and declare Shape class as abstract clas s**

Solution :
option [4] is correct

Q. Consider the following code:

interface MyInterface {

// Method declaration code

}
Which of the following code snippet shows the wrong way to declare Method in interface ?

1). public abstract boolean isValid();

2). public boolean isValid();

3). **protected boolean isValid();**

4). boolean isValid();

Solution :
option [3] is correct

Q. Which of the following statements are true related to interface ?

1). **Interface doesn't alllow to create object .**

2). **Multiple inheritence can be possible in interface.**

3). Class can implement only one interface .

4). Interface can't allow to declare a member variable.

Solution :
option [1,2] are correct

Q. What is the Regular Expression Matching Zero or More Specific Characters?
1). \
2). $
3). *****
4). ^
Solution :
option [3] is correct

Q. public class RegexMatches
```
{
   private static String regex = "dog";
   private static String input = "The dog says meow. " +"All dogs say meow.";
   private static String replace = "cat";

   public static void main(String[] args) {
      //line no.1

     Matcher m = p.matcher(input);
     INPUT = m.replaceAll(replace);
     System.out.println(input);
  }
}
```
Which code need to be inserted at line no.1 to execute it successfully?
1). **Pattern p = Pattern.compile(regex);**
2). Pattern p = Pattern.test(regex);
3). Pattern p = Pattern.matches(regex);
4). Pattern p = new Pattern(regex);
Solution :
option [1] is correct

Q. On which of the given options abstract modifier can be used?
1). constructor
2). static method
3). **non-static methods**
4). **class**

Solution :
option [3,4] are correct

Q. public class Test{
    public static void main(String[] args){
        int[] a = new int[0];
        System.out.print(a.length);
    }
}
1). 1

2). **0**

3). Compilation error, arrays cannot be initialized to zero size.

4). Compilation error, it is a.length() not a.length

Solution :
option [2] is correct

Q. What is the output of the following program?

```
class Test {
public static void main(String[] args) {
try {
doMath(5);
System.out.print("hi");
}
finally { System.out.println(" from finally"); }
}
public static void doMath(int den) {
int num = 7 / den;
}
}
```

1). **hi from finally**

2). hi
from finally

3). prints hi from finally 2 times

4). None of the above

Solution :
option [1] is correct

Q. Given:
try { int number = Integer.parseInt("two"); }

Which could be used to create an appropriate catch block?
1). ClassCastException

2). **IllegalStateException**

3). **NumberFormatException**

4). None of the above is true

Solution :
option [3] is correct

Q. public class Demos {
public static void main(String[] args) {
int arr[5]={1,2,3,4,5};
for(int ele:arr){
System.out.print(str);}
}
What will be the output of above code?
1). 12345

2). no output

3). **ArrayIndexOutOfBounds Exception**

4). **Compilation Error**

Solution :
option [4] is correct

Q.Which of the given statement will ensure that each resource is closed at the end of statement?
1). **try with resource**

2). **call to close() function on the resource**

3). try with resource and finally

4). try with catch

Solution :
option [1] is correct

Q. In Java arrays are _____.
1). **objects**

2). **object references**

3). primitive data type

4). All of the above

Solution :
option [1] is correct

Q._____is raised if I do not provide the String array as the argument to the main method.
1). NullPointerException

2). IllegalAccessException

3). **NoSuchMethodError**

4). **None of the above**

Solution :
option [3] is correct

Q. class ArrayDemo
{
public static void main(String []args){
String str = "Hello World";
int []arr = {1,2,3,4,5};
display(arr,str);
}
public static void display(int …arr,String str)
{
for(int num:arr){System.out.println(num);}
System.out.println(str);
}
}
1). 12345Hello World

2). **Hello World**

3). NumberFormatException

4). **Compilation Error**

Solution :
option [4] is correct

Q. Which of the following statements are true related to exception handling in java ?
1). UserDefined exception can be created by extending from RunTimeException class

2). Throwable is the base class of Error and Exception class .

3). Checked Exception need to be handled either by try and catch block or by using throws keyword i n a code before compilation.

4). **All of the above**

Solution :
option [4] is correct

Q. What will be the output of following program ?

```
public class Foo
{
   public static void main(String[] args)
   {
      try
      {
         return;
      }
      finally
      {
         System.out.println( "Finally" );
      }
   }
}
```

1). **Finally**

2). Compilatation fails .

3). No output.

4). runtime Exception

Solution :
option [1] is correct

Q.

```
class GenericsDemo<T>
{
 T data;
 public GenericsDemo(T data)
 {
  this.data = data;
 }
}
```

Which of the given statement is true about above code?

1). GenericsDemo object can be created by passing any type of parameter

2). GenericsDemo object can be created as given below
GenericsDemo<String>obj = new GenericsDemo<String>();

3). **GenericsDemo object can be created as given below**
GenericsDemo<String>obj = new GenericsDemo<String>("xyz");

4). **Can not create object of given class**

Solution :
option [3] is correct

Q. class CreateObject
{
public static void main(String []args)
{
Set set = new TreeSet();
set.add("Priya");
set.add("Ritu");
set.add(100);
}
}What will happen after the execution of above code?

1). All elements will be successfully added to set

2). last element 100 will not be added to set

3). **ClassCastException**

4). IlleagalStateException

Solution :
option [3] is correct

Q. import java.util.*;
class Test {
public static void main(String[] args) {
// insert code here
obj.add("one");
obj.add("two");
obj.add("TWO");
System.out.println(x.poll());
}
}

Which, inserted at // insert code here, will compile?

1). List<String> obj = new LinkedList<String>();

2). TreeSet<String> obj = new TreeSet<String>();

3). HashSet<String> obj = new HashSet<String>();

4). **Queue<String> obj= new PriorityQueue<String>();**

Solution :
option [4] is correct

Q.A _____ is used to walk through a collection and can remove elements from the collection d uring the iteration.

1). Enumeration

2). **Iterator**

3). ArrayList

4). Vector

Solution :
option [2] is correct

Q. class DemoCmp //line 1
{
int number;
public DemoCmp(int num)
{
number=num;
}
//line 2
}
public class CreateDemo
{
public static void main(String []args){
TreeSet<DemoCmp>set = new TreeSet<DemoCmp>();
set.add(new Demo(8));
set.add(new Demo(2));
set.add(new Demo(3));
}}
Which code need to be inserted at line 1 and line 2
for successful execution of above code?

1). **//line 1**
class DemoCmp implements Comparable

//line 2
public int compareTo(Object obj)
{
return number-obj.number;
}

2). //line 1
class DemoCmp implements Comparator

//line 2
public int compare(Object obj1,Object obj2)
{
return obj1.number-obj2.number;
}

3). //line 1
class DemoCmp implements Comparable

//line 2
public int compare(Object obj1,Object obj2)
{
return obj1.number-obj2.number;
}

4). //line 1
class DemoCmp implements Comparator

//line 2
public int compareTo(Object obj)
{
return number-obj1.number;
}

Solution :
option [1] is correct

Q.Which collection class allows you to grow or shrink its size and provides indexed access to its ele ments, but whose methods are not synchronized?
1). **java.util.HashSet**

2). java.util.Vector

3). **java.util.ArrayList**

4). java.util.List

Solution :
option [3] is correct

Q.A programmer has an algorithm that requires a java.util.List that provides an efficient implementation of add(0,object), but does NOT need to support quick random access.
What supports these requirements?
1). ArrayList

2). Queue

3). Linear List

4). **LinkedList**

Solution :
option [4] is correct

Q. What is the output of the following?

```
import java.util.*;
public class Test {
public static void main(String[] args) {
 Set set = new TreeSet();
 set.add("anu");
 set.add("anil");
 set.add("sunil");
 for(Object str:set){
 System.out.print(str + " ");}
}
}
```

1). **anil anu sunil**

2). anu anil sunil

3). **Compilation error**

4). Exception

Solution :
option [1] is correct

Q. Which of the following methods are declared in List interface ?
1). **get(int index)**

2). **iterator()**

3). **listIterator()**

4). remove()

Solution :
option [1,2] are correct

Q.Which of the given method must be overridden by a class,after implementing Comapartor?
1). **int compare(Object obj);**

2). int compareTo(Object obj);

3). int compareTo(Object obj1,Object obj2);

4). **int compare(Object obj1,Object obj2);**

Solution :
option [4] is correct

Q. Consider the following code snippet:

```
public class Calculator {
       public long add (long number1, long number2){
   return number1 + number2;
}
}

public class CalculatorTest {
     @Test
 public void add() throws Exception {
  Calculator cal=new Calculator();
  assertEquals(4, calculator.add(1,3));
}
}
```

The above code returns an error. What change needs to be done in order to run
the test successfully.
1). There is no error in the code above

2). The TestCase class has not been extended hence it not recognized as a Test class

3). Change the assertEquals method to assertTrue

4). **The assertEqauls method should be written as assertEquals ((long)4,calculator.add(1,3));**

Solution :
option [4] is correct

Q.Which of the following piece of code needs to be inserted so that the program prints the contents of filereader.java?

```
import java.io.*;

class FileReader
{
   public static void main(String args[]) throws Exception
   {
      //Insert code here

      char data[] = new char[1024];
      int charsRead = fileReader.read(data);
      System.out.println(new String(data, 0 , charsRead));

      fileReader.close();
   }
}
```

1). FileInputStream fileReader=new FileInputStream("C:\\myjavacourse\\src\\filereader.java");

2). **FileReader fileReader = new FileReader("C:\\myjavacourse\\src\\filereader.java");**

3). **Both a and b can be used**

4). None of the above is correct

Solution :
option [2] is correct

Q. Which of the following classes in java.io are abstract classes?

1). **Reader**

2). **Writer**

3). **InputStream**

4). File

Solution :
option [1,2,3] are correct

Q. Which of the following I/O classes helps you in persistence storage of Objects?

1). **ObjectOutputStream**

2). **DataOutputStream**

3). FileWriter

4). ObjectWriter

Solution :
option [1] is correct

Q.Which of the following correctly illustrate how an InputStreamReader can be created?
1). **new InputStreamReader(new FileInputStream("data"));**

2). new InputStreamReader(new FileReader("data"));

3). new InputStreamReader("data");

4). **new InputStreamReader(System.in);**

Solution :
option [1,4] are correct

Q.Which of the given annotation is used to do initialization of some task before each test run ?
1). @Test

2). @After

3). @BeforeClass

4). **@Before**

Solution :
option [4] is correct

Q. JUnit is used for _____ in Java Application.
1). Integration Testing

2). Code Review

3). **Unit Testing**

4). Black Box Testing

Solution :
option [3] is correct

Q. class Demo{
public static void main(String []args){
Path javaHome = Paths.get("C:/Program Files/Java/jdk1.8.0_25");
System.out.println(javaHome.getNameCount());}
}What will be the output of above code?
1). 1

2). 2

3). **3**

4). 4

Solution :
option [3] is correct

Q. Which of the following methods of the File class will delete a directory or file?
1). The file class does not allow you to delete a file or directory

2). **remove()**

3). **delete()**

4). deleteFile()

Solution :
option [3] is correct

Q.Which of the given option is correct for adding exception to the test method given below?
Public void divideByZeroTest(){
calobj.divide(15,0);
} Note : calobj is an object of Calculate class which contains divide(int x,int y) method.
1). **@Test(expected = ArithmeticException.class)**

2). @Test(ArithmeticException.class)

3). @Before(ArithmeticException.class)

4). @Before(expected = ArithmeticException.class)

Solution :
option [1] is correct

Q.Which of the given method of Properties class is used to read, key and value pair from input binar y stream?
1). **load(InputStream stream)**

2). **load(Reader reader)**

3). getProperty(String str)

4). getProperty(String str,String default);

Solution :
option [1] is correct

Q. Which of the following methods is best suitable for executing a DML statement?
1). **executeQuery()**

2). **executeUpdate()**

3). execute()

4). getResultSet()

Solution :
option [2] is correct

Q. Which of the following are the interfaces in JDBC API?
1). **DriverManager**

2). **PreparedStatement**

3). **Connection**

4). **ResultSet**

5). **Statement**

Solution :
option [2,3,4,5] are correct

Q. Which of the following methods of ResultSet interface helps in retrieving the type of each column in the resultSet?
1). **getData()**

2). **getMetaData()**

3). getType()

4). getColumnType()

Solution :
option [2] is correct

Q. The scenario where you want to take data from user each time differently, this data you want to pass as an argument to the SQL query. Which of the following statement you will use to execute such query
1). Statement

2). **PreparedStatement**

3). CallableStatement

4). All the above

Solution :
option [2] is correct

Q.ResultSet rs = stmt.executQuery("SELECT name, rank, serialNo FROM employee");
Which of the following will get the value of 'name' from the above ResultSet rs?
1). rs.getString(0)

2). **rs.getString(1)**

3). **rs.getString("name")**

4). None of the above

Solution :
option [2,3] are correct

Q. What is true about Connection Pooling?
1). It increase the performance system

2). It promotes reusability

3). Connection pool implementation not disconnect the link with the database even though client close the connection

4). **All the above**

Solution :
option [4] is correct

Q. Pick up the valid statement to execute the following sql query
PreparedStatement pstmt = con.prepareStatement("insert into student values(?,?));
pstmt.setString(1,"Asha");
pstmt.setString(2,"Basha");
1). **ResutlSet rs = pstmt.executeQuery();**

2). Pstmt.executeQuery();

3). ResultSet rs = pstmt.executeUpdate();

4). **None of the above**

Solution :
option [4] is correct

Q. Which of the following code snippet is most appropriate to roll back the transaction when an error occurs in the transaction?
1).
```
conn.setAutoCommit(false);
        // perform transactions
      conn.commit();
      con.setAutoCommit(true);
 conn .rollback() ;
```

2).
```
try {
      conn.setAutoCommit(false);
        // perform transactions
      conn.commit();
      con.setAutoCommit(true);
 } catch (SQLException e) {
conn .rollback() ;
}
```

3).
```
try {
      conn.setAutoCommit(true);
        // perform transactions
      conn.commit();
      con.setAutoCommit(false);
 } catch (SQLException e) {
conn .rollback() ;
}
```

4). Transaction cannot be rolled back

Solution :
option [2] is correct

Q. Which of the following are pure java drivers ?
1). Type 1 – JDBC-ODBC Bridge

2). Type 2 – Java Native API

3). **Type 3 – Java to Network Protocol**

4). **Type 4 – Java to Database Protocol**

Solution :
option [3,4] are correct

Q. Which of the given statements represents business layer in layered architecture?
1). **Layer in which business methods are implemented**

2). **Business layer implementation is hidden from user only interface is given to user**

3). Business layer implementation is accessible to user

4). Business layer contains implementation for accessing and getting connected with database.

Solution :
option [1,2] are correct

Q. Which of the given is the correct implementation of Test suites?
1). import org.junit.runner.RunWith;
import org.junit.runners.Suite;
@RunWith
@Suite.SuiteClasses({ TestCalAdd.class, TestCalSubtract.class,
TestCalMultiply.class, TestCalDivide.class })
public class CalSuite {

}

2). **import org.junit.runner.RunWith;**
import org.junit.runners.Suite;
@RunWith(Suite.class)
@Suite.SuiteClasses({ TestCalAdd.class, TestCalSubtract.class,
TestCalMultiply.class, TestCalDivide.class })
public class CalSuite {

}

3). import org.junit.runner.RunWith;
import org.junit.runners.Suite;
@Run(Suite.class)
@Suite.SuiteClasses({ TestCalAdd.class, TestCalSubtract.class,
TestCalMultiply.class, TestCalDivide.class })
public class CalSuite {
// the class remains completely empty,
// being used only as a holder for the above annotations

}

4). import org.junit.runner.RunWith;
import org.junit.runners.Suite;

@Suite.SuiteClasses({ TestCalAdd.class, TestCalSubtract.class,
TestCalMultiply.class, TestCalDivide.class })
public class CalSuite {
// the class remains completely empty,
// being used only as a holder for the above annotations
}

Solution :
option [2] is correct

Q.Rima wants to validate user's credential before showing the account details back to user, in online banking application, in which of the given layer she can validate the user's credential?
1). **Presentation Layer**

2). **Business Layer**

3). DataAccessLayer

4). ServiceLayer

Solution :
option [2] is correct

Q. Which of the given statement is true about org.junit.runners.Suite class?
1). **It runs group of test cases**

2). It specifies runner class to run the annotated class

3). **both 0 and 1**

4). none of the above

Solution :
option [1] is correct

Q. Which of the below annotation is used to test with different parameters?
1). **@Parameterized.parameter**

2). **@RunWith(Parameterized.class)**

3). **@Test(Parameterized.class)**

4). @Test.parameter

Solution :
option [1,2] are correct

Q. Choose the correct statements about parameterized test.
1). **It allows to run the same test with different parameters**

2). **It can be specified as given below**
@RunWith(Parameterized.class)

3). **To use it, we need to add static method which returns collection of data**

4). It can be done with @RunWith(Suite.class)

Solution :
option [1,2,3] are correct

Q. Which of the given method is used to create EasyMock?
1). EasyMock.createMock();

2). **EasyMock.createMock(Classname.class);**

3). EasyMock mock = new EasyMock();

4). EasyMock mock = new EasyMock(Classname.class);

Solution :
option [2] is correct

Q. @RunWith(Suite.class)
@Suite.SuiteClasses({ TestPerson.class, TestPerson2.class,TestPersonFixture.class})
public class TestPersonSuite {

```
 @BeforeClass
 public static void setUpBeforeClass() throws Exception {
  System.out.println("Now running the Test Suite");
 }
 @AfterClass
 public static void tearDownAfterClass() throws Exception {
  System.out.println("The Test Suite is completed");
 }
```

}What is true about above code?
1). **It will run three test classes in a group**

2). **@BeforeClass causes the method to be run once before any of the test methods in the cla ss.**

3). **Annotating a public static void method with @AfterClass causes that method to be run aft er all the tests in the class have been run.**

4). @BeforeClass method will run only once at before testing starts.

Solution :
option [1,2,3] are correct

Q. Which of the given options are mock frameworks?
1). **DynaMock**

2). **Jmock**

3). **EasyMock**

4). StaticMock

Solution :
option [1,2,3] are correct

Q. Which of the given statement is true about DTO?
1). **It is used to encapsulate business data**

2). It abstracts the logic to access underlying data store

3). **It is concerned about retrieval,management,processing of application data**

4). It is defined to accept user inputs

Solution :
option [1] is correct

Q. Arrange the priorities in ascending order?
A.INFO B.ERROR c.DEBUG d.FATAL e.WARN
1). **c,a,e,b,d**

2). a,b,c,d,e

3). b,c,d,e,a

4). c,b,d,e,a

Solution :
option [1] is correct

Q. Which of the given class is used to read property file?
1). **Logger**

2). Appender

3). System

4). **PropertyConfigurator**

Solution :
option [4] is correct

Q. What will be the output of the following code?

```
myLogger.setLevel(Level.INFO);
myAppender = new ConsoleAppender(new SimpleLayout());
myLogger.info("Logged since INFO=INFO");
myLogger.debug("Logged, since DEBUG >INFO");
```

1). Logged since INFO=INFO
Logged, since DEBUG >INFO

2). Logged, since DEBUG >INFO

3). Logged since INFO=INFO

4). **No output**

Solution :
option [4] is correct

Q. Log4j can be configured with the help of?
1). XML files

2). **Properties file**

3). **Both a and b is true**

4). Log4j cannot be configured with external files

Solution :
option [3] is correct

Q. Which of the Appender should be used to send LoggingEvent objects to a remote a log server?
1). SysLogAppender

2). ConsoleAppender

3). Telnet Appender

4). **SocketAppender**

Solution :
option [4] is correct

Q. Select the appropriate ways of creating Thread.
1). Extending Thread class

2). Implementing Runnable interface

3). **Both of the above**

4). Only by Extending Thread class

Solution :
option [3] is correct

Q.Which method must be defined by a class implementing the java.lang.Runnable interface?
1). void run()

2). **public void run()**

3). public void start()

4). void run(int priority)

Solution :
option [2] is correct

Q. Which of the following method is responsible to start a Thread?
1). **start()**

2). run()

3). init()

4). go()

Solution :
option [1] is correct
Attempted :

Q. wait(),notify() and notifyAll() methods of _____ class/interface
1). Object

2). **Thread**

3). **Runnable**

4). Class

Solution :
option [2] is correct

Q. Which of the following are methods of Thread class?
1). **isAlive()**

2). **notify()**

3). **wait()**

4). **join()**

Solution :
option [1,4] are correct

Q. Which of the given statements are true about functional interface?
1). **Use of @FunctionalInterface is optional**

2). **It has exactly one abstract method**

3). **It can also have static and default method**

4). It may contain more than one abstract method

Solution :
option [1,2,3] are correct

Q. @FunctionalInterface StringI
{
String getString();
}Which of the given syntax is correct to call this getString() method?
1). ("Welcome to India")->getString();

2). **StringI ref =()->"Welcome to India";**
System.out.println(ref.getString());

3). String ref =("Welcome to India")->getString();
System.out.println(ref);

4). none of the above

Solution :
option [2] is correct

Q. BiFunction<Integer,Integer,Integer>max = (x,y)->(x>y)?x:y;
System.out.println(max.apply(5, 52));
Which of the given statements are true about above code?
1). **It generate the output as 50**

2). **Compilation error, as apply need to be called with 3 arguments**

3). **It will execute successfully if below change is done**
BiFunction<Integer,Integer>max = (x,y)->(x>y)?x:y;

4). **Same expression can also be written as**
BinaryOperator<Integer> maxFunction = (x,y) -> x>y?x:y;
System.out.println(max.apply(5,52));

Solution :
option [1,4] are correct

Q. Which of the given are characteristics of Stream API?
1). **Designed for lamdas**

2). **Can easily be output as an array or list**

3). **Support indexed access**

4). bounded

Solution :
option [1,2] are correct

Q. Which of the given options are built-in functional interfaces?
1). **supplier**

2). **consumer**

3). **predicate**

4). **producer**

Solution :
option [1,2,3] are correct

Q. @FunctionalInterface
interface Power
{
 public int calculatePower(int base,int index);
}
public class CalculatePowerDemo{

 public static void main(String[] args) {
int x=4,y=2;
}
}How to call calculatePower() in CalculatePowerDemo class?

1). **Class CalculatePowerDemo need to implement Power interface and has to override calculatePower() to call it**

2). **It can be implemented as given below**

```
Power pwr = (base,index)->{for(int i=0;i<index;i++)
  {
    base = base*index;
  }
  return base;};
  result= pwr.calculatePower(x, y);
  System.out.println(result);
```

3). It can be implemented as given below

```
Power pwr = (base,index)->{for(int i=0;i<index;i++)
  {
    base = base*index;
  }
  };
  result= pwr.calculatePower(x, y);
  System.out.println(result);
```

4). It can be implemented as given below

```
System.out.println((base,index)->{for(int i=0;i<index;i++)
  {
    base = base*index;
  }
  return base;};)
```

Solution :
option [2] is correct

Q. String []strng = new String[]{"abc","xyz","abcxyz"};
Stream<String>stream = Arrays.stream(strng);
System.out.println(stream.limit(1).findAny());
What will be the output of above code?

1). abc

2). xyz

3). **Optional[abc]**

4). Optional[xyz]

Solution :
option [3] is correct

Q. String []strng = new String[]{"abc","xyz","abcxyz"};
Stream<String>stream = Arrays.stream(strng);
System.out.println(stream.skip(1).findFirst());
What will be the output of above code?
1). abc

2). abcxyz

3). xyz

4). **Optional[xyz]**

Solution :
option [4] is correct

Q. String []strng = new String[]{"abc","xyz","abcxyz"};
Select the correct option to print only the string elements that contains "xyz"?
1). **Stream<String>stream = Arrays.stream(strng);**
stream.filter(str->str.contains("xyz")).forEach(strn->System.out.println(strn));

2). Stream<String>stream = Arrays.stream(strng);
stream.distinct(str->str.contains("xyz")).forEach(strn->System.out.println(strn));

3). System.out.println(stream.allMatch(str1->str1.contains("xyz")));

4). System.out.println(stream.anyMatch(str1->str1.contains("xyz")));

Solution :
option [1] is correct

Q.Which of the given syntax is correct to display elements of an array given below using streams?
String[]strng = new String[]{"IGATE","GLOBAL","SOLUTION"};
1). **Stream<String>stream = Arrays.stream(strng);**
stream.forEach((str)->{
System.out.println(str);});

2). Stream<String>stream = Arrays.stream(strng);
stream.forEach((str)->
System.out.println(str););

3). Stream<String>stream = Arrays.getStream(strng);
stream.forEach((str)->
System.out.println(str););

4). Stream<String>stream = Arrays.getStream(strng);
stream.forEach((str)->{
System.out.println(str);});

Solution :
option [1] is correct