What will be output of following code?                                                                                                    public class Claculator {  
 interface IntegerMath{  
  int operation(int a, int b);  
 }  
   
 public int operationBinary(int a, int b,IntegerMath op) {  
  return op.operation(a,b);  
 }  
 public static void main(String[] args) throws Exception {  
   
  Claculator myapp=new Claculator();  
  IntegerMath addition=(a,b)->a+b;  
  IntegerMath subtraction=(a,b)->a-b;  
  System.out.println("20+10 = "+myapp.operationBinary(20,10, addition));  
  System.out.println("20-10 = "+myapp.operationBinary(20,10, subtraction));  
 }  
}

**1)**. Error at complitime

**2)**. Exception thrown at Runtime

**3)**. **20+10 = 30  
20-10 = 10**

**4)**. Program compiled and run without output

**Solution** :  
option [3] is correct

**Attempted** :  
option [3] is attempted

To execute a stored procedure "total stock" in database server, which of the following code snippet is used?

**1)**. Statement stmt=connection.createStatement(); stmt.execute("totalstatck");

**2)**. **CallableStatementclbstmnt = con.prepareCall(“{call totalStock}”); cs.executeQuery();**

**3)**. **StoreProcedureStatement stmt=connection.createStoreProcedure(“totalStock()”); spstmt.executeQuery();**

**4)**. PrepareStatementpstmt = connection.prepareStatement(“totalStock()”); pstmt.execute();

**Solution** :  
option [2] is correct

**Attempted** :  
option [3] is attempted

**Q.** public class Person{  
  public Person() {  
  System.out.println("Person Object Was Created!!");  
  super();  
  }  
  private String firstName;  
 private String lastName;  
   
 public String getFirstName() {  
  return firstName;  
 }  
 public void setFirstName(String firstName) {  
  this.firstName = firstName;  
 }  
  public static void main(String args[]){  
  Person p1 = new Person();  
  Person p2 = new Person();  
  Person p3 = new Person();    
 }  
}

**1)**. Program will print "Person Object Was Created!!" 3 times.

**2)**. On Execution Program will print "Person Object Was Created!!" Once.

**3)**. Program will throw Exception at runtime

**4)**. **Program will not compile as call to super constructor can only be made in the first line of derived class constructor.**

**Solution** :  
option [4] is correct

**Attempted** :  
option [4] is attempted

**Q.** class D{}  
class T{  
D d;  
public Object getD(){ //line 1  
return new D(); //line 2  
}}  
class S extends T{  
public D getD(){ // line 3  
return new D();  
}}  
Identify the problems in the code?

**1)**. Code will have compilation error at line 1

**2)**. Code will have compilation error at line 2

**3)**. Code will have compilation error at line 3

**4)**. **Code will compile clean**

**Solution** :  
option [4] is correct

**Attempted** :  
option [4] is attempted

**Q.** When is the tearDown() method called in JUnit?

**1)**. After all the tests have run

**2)**. **After each test case has run**

**3)**. At the beginning of every test case

**4)**. At the beginning of the first test case

**Solution** :  
option [2] is correct

**Attempted** :  
option [2] is attempted

**Q.** Which of the following statements are true about Stream API?

**1)**. **Stream Object is constructed Lazily.**

**2)**. **It is used to compute data(Computation on a set of Objects).**

**3)**. **Stream API uses internal iteration to iterate Elements, using forEach methods.**

**4)**. **We can iterate and consume elements from a Stream Object at any number of times**

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

**Attempted** :  
option [2,4] are attempted

**Q.** Which of these does forEach() operates on?

**1)**. **Methods**

**2)**. **Consumer**

**3)**. Producer

**4)**. Predicate

**Solution** :  
option [2] is correct

**Attempted** :  
option [1] is attempted

**Q.**  What is the output of the following code snippet:  
  TreeSet<String> set = new TreeSet<String>();   
  set.add("B");  
  set.add("A");  
  set.add("D");  
  set.add("A");  
  for(String s : set) { System.out.print(s + " " ); }

**1)**. Prints BAD

**2)**. **Prints ABD**

**3)**. Prints: BADA

**4)**. Runtime exception will be thrown

**Solution** :  
option [2] is correct

**Attempted** :  
option [2] is attempted

**Q.** To search for a specified pattern in a column which SQL Keyword can be used ?

**1)**. SQL MATCH

**2)**. **SQL LIKE**

**3)**. SQL PATTERN

**4)**. SQL SEARCH

**Solution** :  
option [2] is correct

**Attempted** :  
option [2] is attempted

**Q.**Which of the following statement(s) with regard to an abstract class in java is/are TRUE?   
A. An abstract class is one that is not used to create objects.   
B.An abstract class is designed only to act as base class to be inheritated by other classes.

**1)**. Only A

**2)**. Only B

**3)**. **Neither A nor B**

**4)**. **Both A and B**

**Solution** :  
option [4] is correct

**Attempted** :  
option [3] is attempted

**Q.** Which type of Statement can execute parameterized queries?

**1)**. Statement

**2)**. **PreparedStatement**

**3)**. ParameterizedStatement

**4)**. **ParameterizedStatement and CallableStatement**

**Solution** :  
option [2] is correct

**Attempted** :  
option [4] is attempted

**Q.** What is the return type of lambda expression?

**1)**. Object

**2)**. String

**3)**. void

**4)**. **Function**

**Solution** :  
option [4] is correct

**Attempted** :  
option [4] is attempted

**Q.** Given:  
   List<Employee> list = new LinkedList<Employee>();  
   What is true for using the below?   
   Collections.sort(list);

**1)**. class Employee should implement Serializable interface

**2)**. class Employee should implement Comparator interface

**3)**. **class Employee should implement Comparable interface**

**4)**. class Employee need not implement any interface

**Solution** :  
option [3] is correct

**Attempted** :  
option [3] is attempted

**Q.**There is exactly one copy of this variable in existence, regardless of how many times the class has been instantiated.  
The above statement is about Which type of variable? 

**1)**. Local variable

**2)**. Global variable

**3)**. Instance variable

**4)**. **Static variable**

**Solution** :  
option [4] is correct

**Attempted** :  
option [4] is attempted

**.** Which of the following methods moves a thread to "running" state?

**1)**. **notify( )**

**2)**. wait()

**3)**. sleep(5000)

**4)**. suspend()

**Solution** :  
option [1] is correct

**Attempted** :  
option [1] is attempted

**Q.** Which of the following are checked type of exceptions?

**1)**. ArithmeticException

**2)**. **InterruptedException**

**3)**. **FileNotFoundException**

**4)**. **SQLException**

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

**Attempted** :  
option [3,4] are attempted

**Q.** Which  rule can be used to test both exception type and message.

**1)**. Catch

**2)**. Expected

**3)**. **ExpectedException**

**4)**. **CatchException**

**Solution** :  
option [3] is correct

**Attempted** :  
option [4] is attempted

**Q.**   
Making the assumptions that:-  
  
1] Connection has been properly initialized  
2] The table MOBILES is valid and exists  
  
Select the correct option for the code give below:-  
    
String insertSQL = "INSERT INTO MOBILES(ID,NAME,PRICE) VALUES(?,?,?)";  
connection.setAutoCommit(false);  
PreparedStatement ps = connection.prepareStatement(insertSQL);  
    
ps.setInt(1, 22);  
ps.setString(2, "SAMSUNG");  
ps.setInt(3, 2900);  
    
ps.executeUpdate();                                                           // LINE-1

**1)**. Code will execute properly and a record will be permanently inserted into the MOBILES table

**2)**. **Code will execute properly but a record will not be permanently inserted into the MOBILES table**

**3)**. **Code will execute properly and a record will be permanently inserted if a line "connection.commit();" is added after the line. marked //LINE-1**

**4)**. Code will execute properly and a record will be permanently inserted if a line "connection.rollback(); is added after the line. marked //LINE-1

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

**Attempted** :  
option [2,3] are attempted

**Q.**Which  class does the err object used in System class belong to as used here? System.err.println("Sample Error Message");

**1)**. **PrintStream**

**2)**. BufferedWriter

**3)**. BufferedOutputStream

**4)**. PrintWriter

**Solution** :  
option [1] is correct

**Attempted** :  
option [1] is attempted

**Q.** Cosider the following code:                            public class SortStringDemo{  
  
public static void main(String args[]){  
  
String[] stringArray={“ef”,”ab”,”cd”};  
  
List<String> stringList=Arrays.asList(stringArray);  
//Line 1  
System.out.println(stringList);  
}  
}  
  
In Line1 which statement if you insert, then the class will print the output in sorted order.

**1)**. **Collections.sort(stringList,(a,b)->a.compareTo(b));**

**2)**. **Collections.sort(stringList,(a,b)->a.compare(b));**

**3)**. Arrays.sort(stringList,(a,b)->a.compareTo(b));

**4)**. Collections.sort(stringList,(a,b)->compareTo(a,b));

**Solution** :  
option [1] is correct

**Attempted** :  
option [2] is attempted

 What will be output of following exception handling program?                                                 import java.io.IOException;  
  
class SuperClass{  
 void method() throws Exception{  
  System.out.println("super-class method");  
 }  
}  
  
class SubClass extends SuperClass{  
 void method() throws IOException{  
  System.out.println("sub-class method");  
 }  
   
}  
public class ExceptionTest {  
  
 public static void main(String[] args) throws Exception {  
 SuperClass obj=new SubClass();  
 obj.method();  
  
 }  
  
}

**1)**. runtime exception

**2)**. compiletime error

**3)**. **super-class method**

**4)**. **sub-class method**

**Solution** :  
option [4] is correct

**Attempted** :  
option [3] is attempted

**Q.** A. establish connection  
B. create Statement object  
C. close the connection  
D. load the driver  
E. create ResultSet object.  
  
Re arrange the steps for connect with the Database

**1)**. **D A B E C**

**2)**. C D A B E

**3)**. A B E C D

**4)**. B D A E C

**Solution** :  
option [1] is correct

**Attempted** :  
option [1] is attempted

**Q.**The \_\_\_\_\_\_\_\_\_\_\_\_\_\_  design pattern decouples method execution from method invocation. Its purpose is to enhance concurrency and simplify synchronized access to objects that reside in their own threads of control

**1)**. **Passive Object**

**2)**. **Active Object**

**3)**. Monitor Object

**4)**. Model Object

**Solution** :  
option [2] is correct

**Attempted** :  
option [1] is attempted

**Q.** Which statement is true concerning the next() method defined in ResultSet class?

**1)**. It will only return a boolean variable indicating if there are more records in the ResultSet

**2)**. **It will only move the current pointer/cursor of the ResultSet to point to the next record in the ResultSet**

**3)**. **It will return a boolean indicating if there are more records in the ResultSet and will also move the current pointer/cursor of the ResultSet to point to the next record in the ResultSet**

**4)**. It fetch the next record from the ResultSet

**Solution** :  
option [3] is correct

**Attempted** :  
option [2] is attempted

**Q.** The import statement in Java

**1)**. includes the source file of the specified classes with the current file before compiling

**2)**. includes the class files of the specified classes with the current class file before after compiling

**3)**. **allows the current code to access classes which are declared in other packages without using their full name.**

**4)**. loads the class files on demand

**Solution** :  
option [3] is correct

**Attempted** :  
option [3] is attempted

**Q.** What will be the output of the following program? public class ExceptionTest {  
  
 public static void main(String[] args) throws Exception {  
 method1();  
 System.out.println("After calling method");  
  
 }  
  
 private static void method1() {  
  method2();  
    
 }  
  
 private static void method2() {  
  method3();  
    
 }  
  
 private static void method3() {  
  throw new NullPointerException();  
    
 }  
  
}

**1)**. **Exception is thrown at runtime**

**2)**. **Error at complietime**

**3)**. After calling method

**4)**. NullponterException After calling method

**Solution** :  
option [1] is correct

**Attempted** :  
option [2] is attempted

**Q.** Which is best suited to a multi-threaded environment?

**1)**. WeakHashMap

**2)**. **Hashtable**

**3)**. HashMap

**4)**. **ConcurrentHashMap**

**Solution** :  
option [4] is correct

**Attempted** :  
option [2] is attempted

**Q.** Given the following code snippet:  
class Parent {  
 protected void test() {  
    
 }  
}  
class Child extends Parent {  
 // Insert test method   
}  
Which  of the following methods can be added to Child Class so that there will not be compilation error and and test () method will remain overriden.

**1)**. void test(int x) { }

**2)**. **public void test( ) {  
  
}**

**3)**. **protected int test( ) {  
  
}**

**4)**. **protected void test( ) { }**

**5)**. **private void test( ) {  
  
}**

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

**Attempted** :  
option [4,5] are attempted

**Q.**Consider the table employee(empId, name, department, salary) and the three queries Q1 ,Q2 and Q3 below. Assuming that department 5 has more than one employee, and we want to find the employees who get higher salary than anyone in the department 5, which one of the statements is TRUE for any arbitrary employee table?                           
Q1 : Select e.empId From employee e  
     Where not exists  
        (Select \* From employee s where s.department = "5" and   
                                        s.salary >=e.salary)  
Q2 : Select e.empId From employee e  
     Where e.salary > Any  
    (Select distinct salary From employee s Where s.department = "5")                                                               Q3 : Select e.empId From employee e  
     Where  exists  
        (Select \* From employee s where s.department = "5" and  e.salary >=s.salary)

**1)**. **Q1 is the correct query**

**2)**. Q2 is the correct query

**3)**. **Q3 is the correct query**

**4)**. Q1 and Q2 is the correct query

**Solution** :  
option [1] is correct

**Attempted** :  
option [3] is attempted

**Q.** What is the result of serializing an Object of SavingsAccount class?  
import java.io.\*;  
class Account {   
  public int balance;    
}  
public class SavingsAccount extends Account{  
  private String name;  
  public static void main(String[] args) throws Exception {  
  SavingsAccount sa = new SavingsAccount();  
  ObjectOutputStream os =   
    new ObjectOutputStream(new FileOutputStream("myFile"));  
  os.writeObject(sa);  
 }  
}

**1)**. **Exception is thrown at runtime**

**2)**. Serializes balance and name

**3)**. Serializes balance only

**4)**. Serializes name only

**5)**. Error at compiletime, class must implement Serializable interface

**Solution** :  
option [1] is correct

**Attempted** :  
No options are Attempted

**Q.** A pipeline is a sequence of what operations in java 8

**1)**. multi-threading

**2)**. concurrent

**3)**. parallel

**4)**. **stream**

**Solution** :  
option [4] is correct

**Attempted** :  
option [4] is attempted

**Q.** Which of the following is true about interfaces in java?

**1)**. **An interface can contain following type of members.   
-public ,static, final fields -default and static methods with bodies**

**2)**. An instance of interface can be created

**3)**. **A class can implement multiple interfaces**

**4)**. **Many classes can implement the same interface**

**Solution** :  
option [1,3,4] are correct

**Attempted** :  
option [3,4] are attempted

**Q.** Which of the following is/are an invalid assignment for the   
Java Collection classes in the java.util package?

**1)**. Collection c1 = new ArrayList();

**2)**. Collection c2 = new HashSet();

**3)**. Collection c3 = new Vector();

**4)**. **Collection c4 = new HashMap();**

**Solution** :  
option [4] is correct

**Attempted** :  
option [4] is attempted

**Q.** Which of the following collection classes can be used to store key value pairs?

**1)**. **TreeMap**

**2)**. TreeSet

**3)**. **Hashtable**

**4)**. **Properties**

**Solution** :  
option [1,3,4] are correct

**Attempted** :  
option [1] is attempted

**Q.** Consider the following code:  
public static void main(String[] args) {   
List cityListing  = null;    
try{  
 cityListing.add("Bangalore");  
 cityListing.add("Mumbai");  
 cityListing.add("Chennai ");  
}  
catch(ArrayIndexOutOfBoundsException ex){  
 cityListing.clear();  
}  
catch(Exception ex){  
 cityListing.clear();  
}  
System.out.println(cityListing.get(2));  
}  
What will be the result of trying to execute  
this program?

**1)**. Compilation Error

**2)**. Program Will Run Successfully and will print:  
Mumbai

**3)**. **Program will throw an Exception when you run it.**

**4)**. **Program Will Run Successfully and will print:  
Bangalore**

**5)**. Program Will Run Successfully and will print:  
Chennai

**Solution** :  
option [3] is correct

**Attempted** :  
option [4] is attempted

**Q.** Which is/are correct statements about primary key of a table?

**1)**. **Primary keys can contain NULL values**

**2)**. **Primary keys cannot contain NULL values**

**3)**. **A table can have only one primary key with single or multiple fields**

**4)**. A table can have multiple primary keys with single or multiple fields

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

**Attempted** :  
option [1,3] are attempted

**Q.** Which collection class is a synchronized one     (Thread Safe class)?

**1)**. ArrayList

**2)**. **Vector**

**3)**. HashSet

**4)**. HashMap

**Solution** :  
option [2] is correct

**Attempted** :  
option [2] is attempted

**.** public class Test {    
public static void meth(boolean b1){  
  System.out.print("Boolean");}  
public static void meth(byte b1){  
  System.out.print("byte ");}  
public static void meth(int i1) {  
  System.out.print("int ");}   
public static void main(String[] args) {     
  byte b1;   
  meth(b1 = 1);  
  meth(b1 == 1);  
}  
}                                                                                        What is the result of attempting to compile and run the program? 

**1)**. **Prints: int Boolean**

**2)**. **Prints: byte Boolean**

**3)**. Prints: Boolean Boolean

**4)**. Compilation error

**Solution** :  
option [2] is correct

**Attempted** :  
option [1] is attempted

**Q.** public class Vehicle {  
Vehicle() {  
 System.out.println(“Create a Vehicle.");  
}  
}  
public class Car extends Vehicle {  
Car() {  
 System.out.println(“Create a Car.");  
}  
}  
public class Parking {  
    public static void main(String[] args) {  
 Car car1 = new Car();  
    }  
}  
  
When execute the program what will be the output?

**1)**. Create a Car. Create a Vehicle

**2)**. **Create a Vehicle.Create a Car.**

**3)**. Create a Vehicle.

**4)**. Create a Car.

**Solution** :  
option [2] is correct

**Attempted** :  
option [2] is attempted

**Q.** Given:                                                                        public class Test{  
 public static void main(String[] args) {  
 int arry[]=new int[] {1,2,3,4,5,6,7,8,9,10};  
 int sum=Arrays.stream(arry).filter(i->i%2==0).sum();  
 System.out.println(sum);  
 //Which of the following expressions will yeild the same result?  
   }                                                                              }

**1)**. **IntStream.rangeClosed(1,10).filter(i->i%2==0).sum();**

**2)**. stream.of(2,3,4,6,8,10).collect(olletors.summingInt(i->i));

**3)**. IntStream.generate(()->10).limit(3).boxed().collect(Collectors.reducing((a,b))->a+b.orElse(0);

**4)**. **Arrays.asList(1,2,3,4,5,6,7,8,9,10).stream().filter(i->i%2==0).reduce(0,(a+b));**

**Solution** :  
option [1] is correct

**Attempted** :  
option [4] is attempted