**Q.** Consider an online shopping site named "shoppee.com".   
Customers are supposed to fill a form for ordering the items.   
What relationship exists between form and fields available in the form?

**1)**. Is - A relationship

**2)**. **Has - A relationship**

**3)**. Realization

**4)**. Generalization

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

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

**Q.** Which access Modifier ensures encapsulation of data?

**1)**. public

**2)**. protected

**3)**. **private**

**4)**. protected internal

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

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

**Q.** Which of the following is not a true UML diagram?

**1)**. Activity Diagram

**2)**. Object Diagram

**3)**. **Deployment Diagram**

**4)**. **Data Flow Diagram**

**5)**. Class Diagram

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

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

**Q.**When two or more classes serve as base class for a derived class the situation is known as \_\_\_\_\_\_\_\_.

**1)**. **Multiple inheritance**

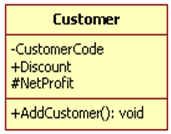
**2)**. Polymorphism

**3)**. Encapsulation

**4)**. Hierarchical inheritance

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

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

**Q.** What will be true as per below diagram?  
  


**1)**. NetProfit attribute is accessible in all classes.

**2)**. Discount attribute is accessible only within the component.

**3)**. **NetProfit attribute is accessible only within the component.**

**4)**. **CustomerCode attribute is not accessible in other class.**

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

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

**Q.** Match the following notations with relevant diagrams:  
  
1. Swim lanes              a. Sequence Diagram  
2. Lifeline                      b. Usecase Diagram  
3. Stick figure               c. Object Diagram  
4. Links                           d. Activity Diagram

**1)**. **1. d  
2. c  
3. b  
4. a**

**2)**. 1. b  
2. c  
3.d  
4. a

**3)**. **1. d  
2. a  
3. b  
4. c**

**4)**. 1. b  
2. d  
3. c  
4. a

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

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

**Q.**Which of the following exception will thrown when a SELECT INTO statement in a PL/SQL block   
returns zero row?

**1)**. invalid\_cursor

**2)**. **no\_data\_found**

**3)**. value\_error

**4)**. too\_many\_rows

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

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

**Q.** Identify the true statement(s) about cursors:  
  
1. PLSQL implicitly declares cursors for all DML statements.  
2. You cannot fetch from a closed cursor.  
3. Fetch statement should be followed by an EXIT condition to avoid infinite looping

**1)**. Only statement 1 is true

**2)**. Only statement 2 is true

**3)**. Only statement 3 is true

**4)**. **All the statements are true**

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

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

**Q.** Consider the below code snippet:  
  
  1  CREATE OR REPLACE PROCEDURE spAddNumber  
  2  (num1 IN NUMBER DEFAULT 10,  
  3  num2 IN NUMBER)  
  4  IS  
  5  total NUMBER;  
  6  BEGIN  
  7     total:=num1+num2;  
  8     DBMS\_OUTPUT.PUT\_LINE('Total: '||total);  
  9  END spAddNumber;  
  
Which statement is valid to be used for executing procedure to generate the below output?  
  
Total: 50

**1)**. **exec spAddNumber(20,30);**

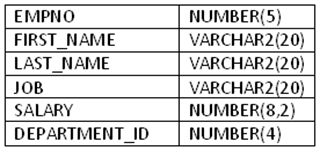
**2)**. exec spAddNumber(num2=>30);

**3)**. **exec spAddNumber(num2=>30,num1=>20);**

**4)**. exec spAddNumber(num2=>30,20);

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

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

**Q.** Examine the structure of EMP table:  
  
  
  
Parvathy wants to view the sorted employee details based on JOB in ascending order.   
If multiple employees are working with same JOB role, then sort the details again with salary   
in descending order. Which of the following statement will match for the requirement?

**1)**. SELECT \* FROM emp ORDERBY salary,job;

**2)**. SELECT \* FROM emp ORDERBY job,salary;

**3)**. **SELECT empno,job,salary FROM emp ORDER BY 2,3 DESC;**

**4)**. **SELECT \* FROM emp ORDERBY job ,salary DESC;**

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

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

**Q.** Consider the below code snippet:  
  
  1  DECLARE  
  2     -- Declaration of Cursor  
  3     CURSOR cname IS SELECT dname FROM dept;  
  4     v\_dname dept.dname%TYPE;  
  5  BEGIN  
  6     FOR rec IN cname  
  7     LOOP  
  8        v\_dname:=rec.dname;  
  9        DBMS\_OUTPUT.PUT\_LINE(v\_dname);  
 10     END LOOP;  
 11     CLOSE cname;  
 12  END;  
  
What will be the output generated if the above procedure gets executed?

**1)**. **Error in Line 11**

**2)**. Error in Line 6

**3)**. "No Data Found" exception occurs if data not exists in dept table.

**4)**. **Displays all the department exists in dept table.**

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

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

**Q.** What will be the output of the following code ?   
  
DECLARE  
row1 emp%ROWTYPE;  
BEGIN  
SELECT \* INTO row1 FROM emp WHERE deptno=99;  
END;

**1)**. **No data found if deptno 99 does not exist**

**2)**. **exact fetch returns more than requested number of rows if there are many employees in dept no 99**

**3)**. **exact fetch returns more than requested number of rows if there is only one employee in dept no 99**

**4)**. compilation error

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

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

**Q.** Assuming there are no employees in deptno 70.  
What does the following code give as output?  
  
DECLARE  
cntr NUMBER;  
myexcep EXCEPTION;  
BEGIN  
SELECT count(\*) INTO cntr FROM emp WHERE deptno=70;  
IF(cntr=0) THEN  
      RAISE myexcep;  
END IF;  
EXCEPTION  
       WHEN NO\_DATA\_FOUND THEN  
              DBMS\_OUTPUT.PUT\_LINE(' no data found');  
       WHEN myexcep THEN  
              DBMS\_OUTPUT.PUT\_LINE(' myexception');  
END;

**1)**. compilation error as myexcep not declared

**2)**. no data found gets printed

**3)**. **myexception gets printed**

**4)**. no output

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

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

**Q.** Consider the following code and predict the output:  
  
DECLARE   
num1 NUMBER;  
BEGIN  
     DBMS\_OUTPUT.PUT\_LINE('Value in num1 is'||num1);  
END;  
/

**1)**. **Value in num1 is**

**2)**. Value in num1 is 0

**3)**. Compilation error as variable num1 is not assigned a value

**4)**. **Value in num1 is some garbage value**

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

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

**Q.** Index is used to:

**1)**. **They are commonly used as read only objects**

**2)**. Increase the row retrieval time.

**3)**. **Reduce disk I/O by locating the data quickly.**

**4)**. Increase the speed of Updates.

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

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

**Q.** If a Procedure is created with the following signature  
Proc1(Emp\_num NUMBER, Emp\_name VARCHAR2 DEFAULT 'Emp name', deptno NUMBER DEFAULT 10)  
then, which of the following option cannot be used to call the above procedure?

**1)**. Proc1(101,'Name1',20)

**2)**. **Proc1(101)**

**3)**. Proc1(101,'Name1')

**4)**. **Proc1**

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

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

**Q.** Which of the following statements are false with respect to XML?  
  
i)   XML document can contain more than one root element.  
ii)  XML tags are case sensitive.  
iii) XML document can contain empty elements.

**1)**. **only i**

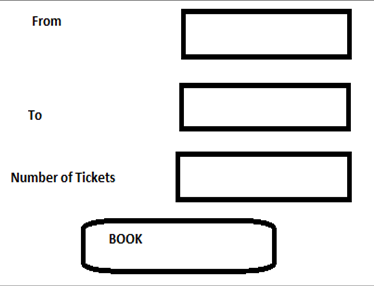
**2)**. i & ii

**3)**. iii & iv

**4)**. ii & iii

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

**Attempted** :  
No options are Attempted

**.** Consider the below form to Book a Rail Ticket.  
  
  
  
The user can book only ten tickets with a transaction. Choose the valid equivalence partitioning for Number of Tickets field in the given scenarion.

**1)**. 100 tickets is Valid.

**2)**. **0 and less than zero is Invalid.**

**3)**. **10 is Valid.**

**4)**. **11 is Invalid.**

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

**Attempted** :  
No options are Attempted

**.** Which of the following are advantages of modular programming?

**1)**. **Modification in any module will not affect our main module.**

**2)**. **The changes made in an appropriate function will affect only that module other modules will   
remain unaffected.**

**3)**. The main module becomes independent of the sub modules and coupling is zeroed.

**4)**. The modules introduce defensive programming technique by minimizing testing effort.

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

**Attempted** :  
No options are Attempted

**Q.** Select the valid statements w.r.t software testing:

**1)**. Software testing is in a way a destructive process

**2)**. Software testing is a process used to help identify the correctness, completeness and   
quality of a developed computer software

**3)**. Software testing is Verifying and Validating if the Software is working as it is intended   
to be working

**4)**. **Software testing is the process of executing a program with the intent of finding the errors**

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

**Attempted** :  
No options are Attempted

**Q.** Consider the below pseudocode:  
                   
BEGIN         
        content[30]= "Pseudocode"  
         FOR i=0 to content[i] not equal to end of character  
               If (content[i]>=97 AND content[i]<=122) Then  
                      content[i]=content[i]-32;  
               END IF   
         END FOR  
END  
  
Which of the good programming practices have been followed in the above code?

**1)**. **Layout**

**2)**. Naming Conventions

**3)**. Modularity

**4)**. Maintainability

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

**Attempted** :  
No options are Attempted

**Q.** Match the following:  
  
A. version  
B. artifactId  
C. groupId  
  
  
1. This is an Id of project's group. This is generally unique amongst an organization or a project.   
For example, a banking group com.company.bank has all bank related projects.  
  
2. This is an Id of the project .This is generally name of the project. For example, consumer-banking.   
Along with the groupId, the artifactId defines the artifact's location within the repository.   
  
3. This is the version of the project.Along with the groupId, It is used within an artifact's repository   
to separate versions from each other.

**1)**. A-1, B-2, C-3

**2)**. **A-3, B-2, C-1**

**3)**. A-2, B-3, C-1

**4)**. A-2, B-1, C-3

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

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

##### Q. \_\_\_\_  is used to create a simple java application in Maven.

### Attempted

1. maven-archetype-java

### Solution

1. maven-archetype-quickstart

2. maven-archetype-jee

3. maven-archetype-mojo

4. maven-archetype-java

**Q.** PMD helps to identify \_\_\_\_\_\_\_\_ and \_\_\_\_\_\_\_\_.

**1)**. **Empty try/catch/finally**

**2)**. **Unnecessary Object creation**

**3)**. Logical Error

**4)**. **Solution to program bugs.**

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

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

**Q.** PMD helps to identify \_\_\_\_\_\_\_\_ and \_\_\_\_\_\_\_\_.

**1)**. **Empty try/catch/finally**

**2)**. **Unnecessary Object creation**

**3)**. Logical Error

**4)**. **Solution to program bugs.**

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

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

**Servlet**

**Q.** Consider the below code snippet exists in pom.xml file  
  
<project>  
  <modelVersion>4.0.0</modelVersion>  
  <groupId>com.cg.bank</groupId>  
  <artifactId>bankapp</artifactId>  
  <version>1.0</version>  
</project>  
  
What type of packaged artifacts will get created when the above mentioned artifacts is packaged   
using "mvn package" command?

**1)**. EAR

**2)**. **WAR**

**3)**. **JAR**

**4)**. SAR

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

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

**Q.** Assumption:  
  
5 web applications are deployed under Jboss application server and each of them   
consists of 2 servlets. How many instances of type ServletContext will be created   
when the server gets started?

**1)**. 10

**2)**. **5**

**3)**. **1**

**4)**. 25

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

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

**Q.**Select the correct statement about the attribute fileSizeThreshold of the annotation @MultipartConfig:

**1)**. Default size is 10 bytes which means files will be immediately written without caching.

**2)**. The fileSizeThreshold is a byte limit above which the file will be saved to cached in RAM.

**3)**. **The file size in bytes after which the file will be temporarily stored on disk. The default size is 0 bytes.**

**4)**. Unless we intend to save all the uploads to the hard drive, you may wish to set this larger than   
the  default to avoid using slower disk -based access when reading the file contents.

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

**Attempted** :  
No options are Attempted

**Q.** @WebServlet("/FileUploadServlet")  
@MultipartConfig(fileSizeThreshold=1024\*1024\*10,    // 10 MB   
                 maxFileSize=1024\*1024\*50,          // 50 MB  
                 maxRequestSize=1024\*1024\*100)      // 100 MB  
public class FileUploadServlet extends HttpServlet {  
  
   protected void doGet(HttpServletRequest request,  
            HttpServletResponse response) throws ServletException, IOException {  
  
             }  
}  
  
Which exception above servlet will generate if we upload the file of size 150MB?

**1)**. FileSizeException

**2)**. FileNotFoundException

**3)**. All of these

**4)**. **IOException**

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

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

**Q.** We have following methods declared in our servlet:  
  
public void service(HttpServletRequest request, HttpServletResponse response)   
{ System.out.println("Service Method"); }  
  
public void doGet(HttpServletRequest request, HttpServletResponse response)   
{ System.out.println("Get Method"); }  
  
public void doPost(HttpServletRequest request, HttpServletResponse response)   
{ System.out.println("Post Method"); }  
This servlet is invoked by an HTML in which we have specified method="post"   
  
What output will it show on the console?

**1)**. **Service Method   
Get Method  
Post Method**

**2)**. Service Method  
Post Method

**3)**. **Service Method**

**4)**. Post Method

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

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

**Q.**In the deployment descriptor, if we have following entries for our servlet "MyServlet"  
  
<servlet>  
 <servlet-name> MyServlet1 </servlet-name>  
 <servlet-class> MyServlet </servlet-class>  
</servlet>  
<servlet-mapping>  
 <servlet-name> MyServlet1 </servlet-name>  
 <url-pattern>/1 </url-pattern>  
 <url-pattern>/2 </url-pattern>  
</servlet-mapping>  
  
Which of the following statements are correct?

**1)**. **It will create one servlet object**

**2)**. It will create two servlet objects

**3)**. **Above entries in deployment descriptor are valid**

**4)**. **Above entries in deployment descriptor are invalid as we can not have multiple   
<url-pattern> tags in <servlet-mapping>**

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

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

**Q.**Following is the code for doGet() method of TestServlet. Which of the given statements   
about it are correct?  
  
public void doGet(HttpServletRequest req, HttpServletResponse res)  
{  
    try  
    {  
        RequestDispatcher rd = this.getServletContext().getRequestDispatcher("Login.jsp"); // 1  
        rd.forward(req, res); // 2  
    }  
    catch(Exception e)  
    {  
        e.printStackTrace();  
    }  
}

**1)**. This will not compile.

**2)**. **This will compile but will give runtime error.**

**3)**. This will compile and work properly.

**4)**. **This will compile. However, it will work properly only if //1 is replaced with:  
RequestDispatcher rd = req.getRequestDispatcher("Login.jsp");**

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

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

 Which of the following statements are TRUE for the code given below?  
  
int MAX\_AGE;   
Cookie cookie = new Cookie("user", user);   
cookie.setMaxAge(MAX\_AGE);   
response.addCookie(cookie);

**1)**. **If MAX\_AGE = 10 the cookie will expire after 10 seconds.**

**2)**. **If MAX\_AGE = 10 the cookie will expire after 600 seconds.**

**3)**. **If MAX\_AGE = 0 the cookie will be deleted.**

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

**Attempted** :  
option [1,2] are attempted

**Q.** @WebServlet("/my-address")  
public class MyServlet extends HttpServlet { ... }  
  
Choose the correct URL.

**1)**. **http://hostname/appname/my-address**

**2)**. **http://hostname/my-address**

**3)**. http://appname/my-address

**4)**. http://hostname/appname

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

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

**Q.** @MultipartConfig defines \_\_\_\_\_\_\_\_\_\_.

**1)**. security constraints

**2)**. Listener

**3)**. **file upload properties**

**4)**. **Servlet**

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

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

**Mod2**

**Q.** Given:  
  
1. class Crivitch {  
2. public static void main(String [] args) {  
3. int x = 0;  
4. // insert code here  
5. do { } while (x++ < y);  
6. System.out.println(x);  
7. }  
8. }  
  
Which statement, inserted at line 4, produces the output 12?

**1)**. int y = 10;

**2)**. **int y = 11;**

**3)**. **int y = 12;**

**4)**. int y = 13;

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

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

What Happens If a JUnit Test Method Is Declared as "private"?

**1)**. Compilation Error

**2)**. **No Error during compliation**

**3)**. **Run Time Error**

**4)**. None of the above

What will be the output of the following code :  
  
class Base  
{  
public void amethod()throws FileNotFoundException{}  
}  
public class ExcepDemo extends Base  
{  
   public static void main(String argv[])  
      {  
         ExcepDemo e = new ExcepDemo();  
      }  
   public void amethod(){}   
   protected ExcepDemo()  
   {  
      try  
       {  
          DataInputStream din = new    
             DataInputStream(System.in);   
          System.out.println("Pausing");   
          din.readByte();  
          System.out.println("Continuing");   
          this.amethod();  
       }  
       catch(IOException ioe) {}  
   }  
}

**1)**. **Compile time error caused by protected constructor**

**2)**. **Compile and run with output of "Pausing" and "Continuing" after a key is hit**

**3)**. Compile time error caused by amethod not declaring Exception

**4)**. Runtime error caused by amethod not declaring Exception

Which relationship enables factoring out infrequent behavior or error conditions?

**1)**. Include

**2)**. **exclude**

**3)**. **extend**

**4)**. intend

@FunctionalInterface  
public interface MyInterface {  
 void checkName();  
 void checkSalary();  
}  
  
Smith has created the above interface, but he is getting some compilation error.  
  
Which of the following options can be used to correct the error?    

**1)**. **Functional Interface should not contain more than one method so one method should be remove   
to resolve compilation error**

**2)**. **Functional Interface should not contain more than one method so if Smith does not want to use this interface   
for Lambda Expression then FunctionalInterface annotation should be removed to resolve the compilation error**

**3)**. Functional Interface should not contain method with void return type so method type should be changed   
to resolve the compilation error

**4)**. Functional Interface must contain a default static method so default static method should be included   
to resolve the compilation error

The Junit framework automatically invokes the \_\_\_\_\_\_\_\_\_\_\_\_ annotated method   
after each test is run.

**1)**. @assertEquals

**2)**. @tearDown

**3)**. **@After**

**4)**. @Before

 Given:  
  
class A {.......}  
class B extends A {.......}  
class C extends A {.......}  
  
What will be the output of the following code fragment'?  
  
A a1 = new B();//Line 1  
A a2 = new C();//Line 2  
B b1 = (B)a1;// Line 3  
B b2 = (B)a2;//Line 4

**1)**. Compile Time Error at Line 1

**2)**. Compile Time Error at Line 3

**3)**. Runtime Exception at Line 3

**4)**. **Runtime Exception at Line 4**

Given:  
  
class A {.......}  
class B extends A {.......}  
class C extends A {.......}  
  
What will be the output of the following code fragment'?  
  
A a1 = new B();//Line 1  
A a2 = new C();//Line 2  
B b1 = (B)a1;// Line 3  
B b2 = (B)a2;//Line 4

**1)**. Compile Time Error at Line 1

**2)**. Compile Time Error at Line 3

**3)**. Runtime Exception at Line 3

**4)**. **Runtime Exception at Line 4**

Which annotation can be used to carry out cleanup activities and to execute the method   
after all tests have finished ?

**1)**. **@After**

**2)**. @Before

**3)**. **@AfterClass**

**4)**. @BeforeClass

Suppose you want to test object response for not null.   
Which of the following code would you use in a test to verify than an object is not null?

**1)**. **assertNotNull("Response is null", response)**

**2)**. **assertNotNull(response)**

**3)**. if(response = = null) {throw newExcpetion("The response is null");}

**4)**. assertNull("Response is null", response)

Which of the following line of code is suitable to start a thread ?  
  
class X implements Runnable   
{   
    public static void main(String args[])   
    {  
        /\* Missing code? \*/  
    }   
    public void run() {}   
}

**1)**. Thread thread = new Thread(MyThread);

**2)**. Thread t = new Thread(MyThread);  
t.start();

**3)**. **MyThread run = new MyThread(); Thread t = new Thread(run); t.start();**

**4)**. **Thread thread = new Thread();   
thread.run();**

Select the false statement about Lambda Expressions:

**1)**. **Increases the Lines of code**

**2)**. Provides sequential and parallel execution support

**3)**. Allows to pass behaviors into methods

**4)**. Higher efficiency with laziness

Assuming a method contains code which may raise an Exception (but not a RuntimeException),   
What is the correct way for a method to indicate that it expects the caller to handle that exception:

**1)**. throw Exception

**2)**. **throws Exception**

**3)**. new Exception

**4)**. No need to specify anything

When testing an error flow, what test would you write?

**1)**. **public void testErrorCase() throws Exception  
     {  
     methodToTest();   
     // Asserts here }**

**2)**. public void testErrorCase(){ try{   
     methodToTest();   
     }  
     catch (Exception expected)  
     {   
     // Asserts her   
     }   
     }

**3)**. **public void testErrorCase(){   
     try{   
     methodToTest();    
     fail("Should not reach this point")  
     catch (Exception expected)  
     {    
     // Asserts here   
     }   
     }**

What is the result of trying to compile and run the following program?  
  
 class  Phone implements Cloneable{  
public static void main(String[] args) {  
  Phone p= new Phone();  
  if (p instanceof Object)  
     System.out.println("Object");  
if (p instanceof Cloneable)  
     System.out.println("Cloneable");  
}  
}

**1)**. The program does not compile

**2)**. The program compiles and runs and writes Object to the standard ouput

**3)**. **The program compiles and runs and writes both Object and Cloneable to the standard ouput.**

**4)**. The program compiles and runs and writes Cloneable to the standard ouput.

class ArrayTest {  
    public static void main(String args[]) {  
      int arr[] = new int[2];  
      System.out.println(arr[0]);  
    }  
}  
  
What will be the result of compiling and executing the above code?

**1)**. The program does not compile because arr[0] is being read before being initialized.

**2)**. The program generates a runtime exception because arr[0] is being read before being initialized.

**3)**. **The program compiles and prints 0 when executed**

**4)**. The program compiles and prints 1 when executed.

**5)**. The program compiles and runs but the results are not predictable because of un-initialized   
memory being read

Consider the below code of an anonymous class:  
  
Comparator<Integer> intComparator = new Comparator<Integer>() {  
       @Override  
       public int compare(Integer o1, Integer o2) {  
              return o1.compareTo(o2);  
       }  
};  
Which of the following are valid lambda expressions to replace this anonymous class?

**1)**. **(o1,o2) -> o1.compareTo(o2)**

**2)**. **(o1,o2)->return o1.compareTo(o2)**

**3)**. **(Integer o1, Integer o2)->o1.compareTo(o2)**

**4)**. **(int o1, int o2)->o1.compareTo(o2)**

What is the output of the following program?  
  
class A {  
  private void print() {   
    System.out.println("Inside A's print method.");   
  }     
  public void test() {   
    this.print();   
  }   
}   
class B extends A {   
  private void print() {   
    System.out.println("Inside B's print method.");   
  }             
  public static void main(String args[]) {   
    A a = new B();   
    a.test();   
  }   
}

**1)**. **Inside A's print method.**

**2)**. Inside B's print method.

**3)**. Inside A's print method.   
     Inside B's print method.

**4)**. **Compilation error**

Read the following example. Which of the following options relates to this example?  
  
"Customer needs to know what is the interest he is earning; and may not need to know how the bank is calculating this interest."

**1)**. Encapsulation

**2)**. Inheritance

**3)**. Polymorphism

**4)**. **Abstraction**

 Given:  
  
class CardBoard {  
Short story = 5;  
CardBoard go(CardBoard cb) {  
cb = null;  
return cb;  
}  
public static void main(String[] args) {  
CardBoard c1 = new CardBoard();  
CardBoard c2 = new CardBoard();  
CardBoard c3 = c1.go(c2);  
c1 = null;  
// do Stuff  
} }  
  
When // doStuff is reached, how many CardBoard objects are eligible for GC?

**1)**. 0

**2)**. **1**

**3)**. **2**

**4)**. None of the above

 public class MethodOverriding  
{  
  private void add(int operand1, int operand2)  
  {  
     System.out.println(operand1 + operand2);  
  }  
}  
public class Overridden extends MethodOverriding  
{  
  public void show()  
  {  
     add(10, 12);  
  }  
   public static void main(String args[])  
   {  
      Overridden ob = new Overridden();  
      ob.show();  
   }  
}  
  
What will be the output of above code when compiled and executed?

**1)**. **Compile time error**

**2)**. Runtime error as add method is not defined in MethodOverriding class

**3)**. Will compile and display 32

**4)**. Will compile and dislay 1012

##### Saratha is involved in the development of a Car sales purchase application.  She created different packages named as com.carsales.service, com.carsales.dao, com.carsales.dto  and com.carsales.mainclass. She created a class named "CarSales" in a package " com.carsales.dto"  which consists of attributes such as carId, carName, and price and appropriate setters and getters.  She also created a class named "CarSalesDAO" in a package " com.carsales.dao".  Implementation of which statement allows to access attributes of "CarSales"  class from a  "com.carsales.dto" package in a "CarSalesDAO" class available in a  "com.carsales.dao" package?

**1)**. Assigning attributes and a class with public.

**2)**. Assigning private modifier to attributes and public modifier to class and also inheriting  CarSales class in   
CarSalesDAO class will helpful for performing this task.

**3)**. Assigning public modifier only to attributes.

**4)**. **Assigning public modifier to class and private modifier to attributes and also importing CarSales class   
in CarSalesDAO class will helpful for performing this task.**

 Consider the following entries in the log4j property file:  
  
log4j.appender.testAppender=org.apache.log4j.ConsoleAppender  
log4j.appender.testAppender.layout=org.apache.log4j.PatternLayout  
log4j.appender.testAppender.layout.conversionPattern=%m%n  
  
Which of the following statements is true about the above entries?

**1)**. **set the appender named testAppender to be a console appender**

**2)**. **set the name of Appender to ConsoleAppender**

**3)**. **the layout of ConsoleAppender is set to PatternLayout**

**4)**. **set the layout for the appender testAppender to PatternLayout**

Choose the correct lambda code to check the given value is divisible by 5.

**1)**. Predicate<Integer> divisble=(number)->number / 5 == 0;  
 System.out.println(divisble.test(11));

**2)**. Predicate<int> divisble=(number)->number / 5 == 0;  
 System.out.println(divisble.test(11));

**3)**. **Predicate<Integer> divisble=(number)->number % 5 == 0;  
 System.out.println(divisble.test(11));**

**4)**. Predicate<int> divisble=(number)->number % 5 == 0;  
 System.out.println(divisble.test(11));

Integer i = new Integer(10);  
int i2 = 10;  
System.out.println( i == i2 );  
  
What will be the output of the above code fragment?

**1)**. **Will print true**

**2)**. **Compilation Error**

**3)**. CastException

**4)**. Runtime Exception

**5)**. Will print false

What will be the output of the following code fragment?  
  
Vector v1 = new Vector(7, 3);  
 for(int num = 1;num<=15;num++)  
  v1.add(num);  
 System.out.println(v1.capacity());

**1)**. 21

**2)**. **15**

**3)**. 18

**4)**. **16**

If Single files are to be read one after the other simultaneously, the appropriate class to be used is:

**1)**. **SequenceInputStream**

**2)**. RandomAccessFile

**3)**. FileReader

**4)**. **FileInputStream**

**5)**. DataInputStream

Given:  
  
class Alpha{  
   Alpha doStuff(char c){  
     return new Alpha();  
   }  
}  
  
class Beta extends Alpha{  
  Beta doStuff(char c){  
    return new Beta();  
  }  
}  
  
Is it a valid overriding?

**1)**. **True**

**2)**. **False**

Here is log4j.properties file:  
  
1. log4j.rootLogger=ERROR, CA  
2. log4j.appender.CA = org.apache.log4j.ConsoleAppender  
3. log4j.appender.CA.layout=org.apache.log4j.HTMLLayout  
  
What will be the result of attempting to compile and run the following program ?  
  
public static void main(String[] args) {  
  PropertyConfigurator.configure("log4j.properties");  
logger.debug("Here goes the debug message");  
logger.error("Here goes the error message");  
logger.fatal("Here goes the fatal message");  
logger.info("Here goes the info message");  
logger.warn("Here goes the warn message");  
   }  
}

**1)**. Here goes the debug message  
Here goes the error message  
Here goes the fatal message  
Here goes the info message  
Here goes the warn message

**2)**. **Here goes the debug message  
Here goes the error message  
Here goes the fatal message**

**3)**. Here goes the error message  
Here goes the fatal message  
Here goes the info message

**4)**. **Here goes the error message  
Here goes the fatal message**

**5)**. Here goes the info message  
Here goes the error message  
Here goes the fatal message

**6)**. Here goes the info message  
Here goes the warn message  
Here goes the error message  
Here goes the fatal message

A system integrator needs to know the scalability of the system. Which UML building block will be helpful   
for examining the same?

**1)**. Structural view

**2)**. **Behavioral View**

**3)**. Implementation View

**4)**. **Environment View**

import java.util.TreeSet;  
public class Employee {  
  private int empno;  
  private String name;  
  
  public static void main(String args[]) {  
    TreeSet empset = new TreeSet();  
    empset.add(new Employee());  
    empset.add(new Employee());  
    System.out.println("Size: " + empset.size());  
  }  
}

**1)**. Compilation Error

**2)**. **Runtime Exception**

**3)**. Size: 1

**4)**. **Size: 2**

"Books has pages" is an example of \_\_\_\_\_\_\_\_\_\_.

**1)**. Generalization

**2)**. **composition**

**3)**. aggregation

**4)**. None of the above

 "Library contains books" is an example of \_\_\_\_\_\_\_\_\_\_\_\_.

**1)**. **Generalization**

**2)**. composition

**3)**. **aggregation**

**4)**. None of the above

What will be the output of the following code fragment?  
  
HashMap map = new HashMap();  
  map.put("obj1", null);  
  map.put("obj2", null);  
  map.put(null, null);  
  map.put("obj3", "obj3");  
  System.out.println(map.size());

**1)**. 3

**2)**. 1

**3)**. **4**

**4)**. 2

Given:  
  
class Mixer {  
Mixer() { }  
Mixer(Mixer m) { m1 = m; }  
Mixer m1;  
public static void main(String[] args) {  
Mixer m2 = new Mixer();  
Mixer m3 = new Mixer(m2); m3.go();  
Mixer m4 = m3.m1; m4.go();  
Mixer m5 = m2.m1; m5.go();  
}  
void go() { System.out.print("hi "); }  
}  
  
What is the result?

**1)**. hi hi

**2)**. hi hi hi

**3)**. **Compilation fails**

**4)**. **hi hi, followed by an exception**

**Q.**Consider the following code which is used to print a list of integers stored in "numbers" list.   
  
numbers.forEach((Integer value) -> System.out.println(value));  
  
Which one of the following built-in interface referenced by above lambda expression?

**1)**. Function<Integer>

**2)**. **Consumer<Integer>**

**3)**. Supplier<Integer>

**4)**. **Predicate<Integer>**

A setUp() and tearDown() code can be run once for all of the tests by:

**1)**. **Adding a @BeforeClass annotation to a method to be run before all tests in a class**

**2)**. **Adding a @AfterClass annotation to a method to be run after all tests in a class**

**3)**. This can not be done

**4)**. setUp() and tearDown() will be called only once for all the tests

 Given:  
  
public interface SuperInterface {....................}  
public interface SubInterface extends SuperInterface {...........}  
public class MyClass implements SuperInterface {.............}  
MyClass obj = new MyClass();  
  
With how many types, reference obj is compatible with including the default type "Object"?

**1)**. 0

**2)**. 1

**3)**. 2

**4)**. **3**

**Q.**Which method puts the current thread on wait until the thread on which it's called is dead?

**1)**. sleep()

**2)**. **join()**

**3)**. start()

**4)**. stop()

**Q.**Which diagrams define the static architecture of a model. They are used to model the “things”   
that make up a model. They are used to model the relationships and dependencies?

**1)**. **Structural view**

**2)**. Behavioral view

**3)**. Implementation view

**4)**. Environment view

**Q.**\_\_\_\_\_\_ is the relationship between an “interface” and the “class” that provides the interface's services.

**1)**. Visualization

**2)**. **Realization**

**3)**. association

**4)**. composite