

Assume that `library` represents a list of objects of type `Book`. What does the variable `count` represent at the end of execution of this code?

```
1 # assume that library has n elements, only three of which are shown below
2 Book.count = 0
3 library = [Book('A Brief History of Time', 250, 'Hawking', 'Non Fiction'),
4            Book('Alice in Wonderland', 300, 'Carroll', 'Fiction'),
5            Book('Feynman Lectures in Physics', 400, 'Feynman', 'Non Fiction')]
6
7 count = 0
8 for book in library:
9     if book.is_fiction():
10         count = count + 1
```

Options :

6406531884606. ✔ Number of fiction books in the library

6406531884607. ✖ Number of non-fiction books in the library

6406531884608. ✖ Number of books in the library

6406531884609. ✖ None of these

Java

Section Id :	64065338321
Section Number :	4
Section type :	Online
Mandatory or Optional :	Mandatory
Number of Questions :	24
Number of Questions to be attempted :	24
Section Marks :	100
Display Number Panel :	Yes
Group All Questions :	No
Enable Mark as Answered Mark for Review and Clear Response :	Yes

Maximum Instruction Time : 0

Sub-Section Number : 1

Sub-Section Id : 64065380367

Question Shuffling Allowed : No

Is Section Default? : null

Question Number : 75 Question Id : 640653563781 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 0

Question Label : Multiple Choice Question

THIS IS QUESTION PAPER FOR THE SUBJECT "DIPLOMA LEVEL : PROGRAMMING CONCEPTS USING JAVA (COMPUTER BASED EXAM)"

ARE YOU SURE YOU HAVE TO WRITE EXAM FOR THIS SUBJECT?

CROSS CHECK YOUR HALL TICKET TO CONFIRM THE SUBJECTS TO BE WRITTEN.

(IF IT IS NOT THE CORRECT SUBJECT, PLS CHECK THE SECTION AT THE [TOP](#) FOR THE SUBJECTS REGISTERED BY YOU)

Options :

6406531884610.  YES

6406531884611.  NO

Sub-Section Number : 2

Sub-Section Id : 64065380368

Question Shuffling Allowed : Yes

Is Section Default? : null

Question Number : 76 Question Id : 640653563782 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 4

Question Label : Multiple Choice Question

Consider the Java code given below.

```
1  class ClassOne{
2      public void methodOne(){
3          // ...
4      }
5      public void methodTwo(){
6          // ...
7          methodOne();
8          // ...
9      }
10 }
11 class ClassTwo{
12     public static void methodThree(){
13         // ...
14         ClassOne c = new ClassOne();
15         c.methodTwo();
16         // ...
17     }
18     public static void main(String[] args) {
19         // ...
20         methodThree();
21     }
22 }
```

During execution of Line 8 in the above code, the activation record of which method is at the top of the stack of activation records?

Options :

6406531884612. ✖ main

6406531884613. ✖ methodOne

6406531884614. ✔ methodTwo

6406531884615. ✖ methodThree

Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 4

Question Label : Multiple Choice Question

Consider the following code.

```
class Equipment implements Cloneable{
    String type;
    // Constructor
    // Accessor method getType()
    // Mutator method setType()
    public Equipment clone() throws CloneNotSupportedException{
        return (Equipment)super.clone();
    }
}
class Lab implements Cloneable{
    Equipment eqp;
    String name;
    // Constructor
    // Mutator method setName()
    public Lab clone() throws CloneNotSupportedException{
        Lab lb = (Lab)super.clone();
        lb.eqp = lb.eqp.clone();
        return lb;
    }
    public String toString(){
        return eqp.getType() + ":" + name;
    }
}
public class Test {
    public static void main(String[] args) {
        Lab lb1 = new Lab(new Equipment("Computer"), "Computer");
        try{
            Lab lb2 = lb1.clone();
            lb2.eqp.setType("Milling Machine");
            lb2.setName("Mechanical");
            System.out.println(lb1);
            System.out.println(lb2);
        }
        catch(Exception e){
            System.out.println(e);
        }
    }
}
```

What will the output be?

Options :

6406531884616. ✖ Milling Machine:Computer
Milling Machine:Mechanical

6406531884617. ✖ Milling Machine:Mechanical
Milling Machine:Mechanical

6406531884618. ✔ Computer:Computer
Milling Machine:Mechanical

6406531884619. ✖ Computer:Mechanical
Milling Machine:Mechanical

Question Number : 78 Question Id : 640653563785 Question Type : MCQ Is Question

Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 4

Question Label : Multiple Choice Question

Consider the code given below

```

class Animal{
    public void legs(){
        System.out.println("Number of legs are not known");
    }
    public void wings(){
        System.out.println("Wings may exist");
    }
}
class Dog extends Animal{
    public void legs(){
        System.out.println("Four legs");
    }
}
class Ostrich extends Animal{
    public void legs(){
        System.out.println("Two legs");
    }
    public void wings(){
        System.out.println("Two wings");
    }
}
public class Test{
    static void show(Animal[] a){
        for(int i = 0; i < a.length; i++){
            a[i].legs();
            a[i].wings();
        }
    }
    public static void main(String[] args){
        Animal[] a = {new Dog(), new Ostrich()}; // LINE 1
        show(a);
    }
}

```

Choose the correct option.

Options :

This code generates the output:

Two legs
Two wings
Two legs
Two wings

6406531884624. ✖

6406531884625. ✔

This code generates the output:

```
Four legs  
Wings may exist  
Two legs  
Two wings
```

This code generates the output:

```
Number of legs are not known  
Wings may exist  
Number of legs are not known  
Wings may exist
```

6406531884626. ✖

Compilation error at LINE 1 because a reference variable of type Animal cannot refer to an object of class Ostrich

6406531884627. ✖

Question Number : 79 Question Id : 640653563786 Question Type : MCQ Is Question

Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction

Time : 0

Correct Marks : 4

Question Label : Multiple Choice Question

Consider the code given below that checks whether two rectangles are the same. Method `equals` is overridden to compare two `Rectangle` objects as follows: If two rectangles have the same area, then they are the same. Based on the given information, answer the question that follows.

```
class Rectangle{
    private int length;
    private int breadth;

    //Constructor to initialize instance variables

    public double area() {
        return (length * breadth);
    }
    public boolean equals(Object obj) {
        // CODE BLOCK
    }
}

public class Test {
    public static void main(String[] args) {
        Rectangle r1 = new Rectangle(3,5);
        Rectangle r2 = new Rectangle(5,3);
        if(r1.equals(r2))
            System.out.println("r1 and r2 are the same");
        else
            System.out.println("r1 and r2 are not same");
    }
}
```

Choose the correct option to fill in place of CODE BLOCK so that the output is:

r1 and r2 are the same

Options :

```
if(obj instanceof Rectangle) {
    Rectangle r = (Rectangle) obj;
    if(this.area() == r.area())
        return false;
}
```

6406531884628. ✖ return true;

6406531884629. ✖


```
if(obj instanceof Rectangle) {  
    Rectangle r = obj;  
    if(this.area() == r.area())  
        return true;  
}  
return false;
```

```
if(obj instanceof Rectangle) {  
    Rectangle r = obj;  
    if(this.area() == r.area())  
        return false;  
}
```

6406531884630. ✖ return true;

```
if(obj instanceof Rectangle) {  
    Rectangle r = (Rectangle) obj;  
    if(this.area() == r.area())  
        return true;  
}
```

6406531884631. ✔ return false;

Question Number : 80 Question Id : 640653563790 Question Type : MCQ Is Question

Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 4

Question Label : Multiple Choice Question

Consider the Java code given below, and answer the question that follows:

```

abstract class Op implements Runnable{
    static int sum = 0;
}
class Op1 extends Op{
    public void run() {
        if (sum != 4) {
            sum = sum + 5;
        }
    }
}
class Op2 extends Op{
    public void run() {
        if (sum != 5) {
            sum = sum + 4;
        }
    }
}
public class RaceCondition {
    public static void main(String[] args) {
        Op o1 = new Op1();
        Op o2 = new Op2();

        Thread t1 = new Thread(o1);
        Thread t2 = new Thread(o2);

        t1.start();
        t2.start();

        System.out.println(Op.sum);
    }
}

```

Choose the correct option.

Options :

- 6406531884644. ✖ The program will never generate the output: 0
- 6406531884645. ✖ The program will never generate the output: 4
- 6406531884646. ✖ The program will never generate the output: 5
- 6406531884647. ✖ The program will never generate the output: 9
- 6406531884648. ✔ The output can be 0 or 4 or 5 or 9.

Question Number : 81 Question Id : 640653563792 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 4

Question Label : Multiple Choice Question

Consider the code given below.

```
class Circle{
    static final double PI = 3.14;
    double radius;
    public Circle(double r){
        radius = r;
    }
    public Circle(Circle c){
        radius = c.radius;
    }
    public double perimeter(){
        return 2 * PI * radius;
    }
}

public class ConTest {
    public static void main(String[] args) {
        Circle c1 = new Circle(6.0);
        Circle c2 = new Circle(c1);
        Circle c3 = new Circle(c2);
        c1.radius = 4.0;
        System.out.println(c1.perimeter());
        System.out.println(c2.perimeter());
        System.out.println(c3.perimeter());
    }
}
```

What will the output be?

Options :

25.12
25.12
6406531884653. ✖ 25.12

37.68
37.68
6406531884654. ✖ 37.68

25.12
37.68
6406531884655. ✓ 37.68

37.68
25.12
6406531884656. ✖ 25.12

Question Number : 82 Question Id : 640653563796 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 4

Question Label : Multiple Choice Question

Consider the code given below.

```
import java.util.*;
public class MapTest {
    public static void main(String[] args) {
        var batsmen= new LinkedHashMap<String,Integer>();
        batsmen.put("Sachin", 100);
        batsmen.put("Sachin", 55 + batsmen.getOrDefault("Sachin", 0));
        batsmen.put("Sehwag", 100);
        batsmen.put("Sehwag", 55);
        for(Map.Entry<String, Integer> obj1:batsmen.entrySet())
            System.out.println(obj1.getKey()+":"+obj1.getValue());
    }
}
```

Choose the correct option.

Options :

This program generates the output:
Sachin:55
Sehwag:55
6406531884669. ✖

This program generates the output:

Sachin:155

6406531884670. ✖ Sehwag:155

This program generates the output:

Sachin:100

6406531884671. ✖ Sehwag:100

This program generates the output:

Sachin:155

6406531884672. ✔ Sehwag:55

Question Number : 83 Question Id : 640653563797 Question Type : MCQ Is Question

Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 4

Question Label : Multiple Choice Question

Consider the Java code given below.

```

class RightAngledException extends Exception{
    public RightAngledException(String str) {
        super(str);
    }
}

class Triangle{
    double hypotenuse, base, altitude;
    //Constructor to initialize instance variables
    public boolean isRightAngled() throws RightAngledException{
        double hyp = hypotenuse * hypotenuse;
        double b = base * base;
        double a = altitude * altitude;
        if(hyp != (b+a))
            throw new RightAngledException("Not a Right Angled Triangle");
        else
            return true;
    }
}

public class ExceptionTest {
    public static void main(String[] args) {
        var obj1 = new Triangle(13, 12, 10);
        var obj2 = new Triangle(5, 4, 3);
        try {
            System.out.println(obj1.isRightAngled());
            System.out.println(obj2.isRightAngled());
        } catch (RightAngledException e) {
            System.out.println(e.getMessage());
        }
    }
}

```

What will the output be?

Options :

6406531884673. ✖ Not a Right Angled Triangle
Not a Right Angled Triangle

6406531884674. ✔ Not a Right Angled Triangle

6406531884675. ✖ true
Not a Right Angled Triangle

6406531884676. ✖ true

Question Number : 84 Question Id : 640653563798 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 4

Question Label : Multiple Choice Question

Consider the Java code given below.

```
import java.io.*;
class CreditCard implements Serializable{
    private String cardNo = "*****";
    private transient int pin = 1000;
    private String exp = "00/00";
    public CreditCard(String cno, int p, String e) {
        cardNo = cno;
        pin = p;
        exp = e;
    }
    public String toString() {
        return cardNo + ", " + pin + ", " + exp;
    }
}
public class SerialTest{
    public static void main(String[] args) throws Exception{
        var fos = new FileOutputStream("credit.txt");
        var os = new ObjectOutputStream(fos);
        os.writeObject(new CreditCard("4688171329130605", 9999, "03/23"));

        var fis = new FileInputStream("credit.txt");
        var ois = new ObjectInputStream(fis);
        CreditCard card = (CreditCard)ois.readObject();
        System.out.println(card);
    }
}
```

What will the output be?

Options :

6406531884677. ✖ null, 0, null

6406531884678. ✖ 4688171329130605, 1000, 03/23

6406531884679. ✖ 4688171329130605, 9999, 03/23

6406531884680. ✔ 4688171329130605, 0, 03/23

6406531884681. ✖ *****, 1000, 00/00

Question Number : 85 Question Id : 640653563799 Question Type : MCQ Is Question

Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 4

Question Label : Multiple Choice Question

Consider the code given below.

```

import java.util.*;
class Team{
    HashMap<String, String> f = new HashMap<String, String>();
    public Team() {
        f.put("CSK", "Radhakrishnan");
        f.put("MI", "Ambani");
    }
    public String getOwner(String t){
        return f.get(t);
    }
}
public class OptionalTest {
    public static void main(String[] args){
        Optional<String> op1 = Optional.ofNullable(new Team().getOwner("CSK"));
        Optional<String> op2 = Optional.ofNullable(new Team().getOwner("RCB"));
        op1.ifPresent(n->System.out.println(n.toUpperCase()));
        op2.ifPresent(n->System.out.println(n.toUpperCase()));
    }
}

```

Choose the correct option.

Options :

6406531884682. ✓ This program generates the output:
RADHAKRISHNAN
6406531884683. ✗ This program terminates due to NullPointerException after printing the message:
RADHAKRISHNAN
6406531884684. ✗ This program generates the output:
RADHAKRISHNAN
null
6406531884685. ✗ This program generates no output.

Question Number : 86 Question Id : 640653563800 Question Type : MCQ Is Question

Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction

Time : 0

Correct Marks : 4

Question Label : Multiple Choice Question

Consider the code given below.

```
import java.util.*;
import java.util.stream.*;
class Faculty{
    String name;
    double salary;
    //Constructor to initialize instance variables
    //Method toString() to return name of the faculty
}
public class CollectingTest {
    public static void main(String[] args){
        var fArr = new ArrayList<Faculty>();
        fArr.add(new Faculty("Sravya",30000.00));
        fArr.add(new Faculty("Thanvi",50000.00));
        fArr.add(new Faculty("Sharadha",100000.00));
        fArr.add(new Faculty("Pooja",29000.00));
        Map<Boolean, List<Faculty>> facMap;
        facMap = fArr.stream()
            .collect(Collectors.partitioningBy(f->f.salary >= 50000.00));
        System.out.println(facMap.get(false));
    }
}
```

Choose the correct option.

Options :

6406531884686. ✖ This program generates the output:
[Thanvi, Sharadha]

6406531884687. ✖ This program generates the output:
[Sharadha]

6406531884688. ✔ This program generates the output:
[Sravya, Pooja]

6406531884689. ✖

This program generates the output:
[Sravya, Thanvi, Pooja]

Question Number : 87 Question Id : 640653563801 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 4

Question Label : Multiple Choice Question

Consider the code given below.

Assume that file "abc.txt" exists and contains the following text.

Hello IITM students

Assume that there is no file named "xyz.txt".

```
import java.io.*;
import java.util.Scanner;
public class FileTest {
    public static void main(String[] args) throws Exception{
        try {
            FileInputStream in = new FileInputStream("abc.txt");
            Scanner sc = new Scanner(in);
            String data = "";
            if(sc.hasNext())
                data = sc.nextLine();

            var out = new FileOutputStream("xyz.txt");
            var dout = new DataOutputStream(out);
            dout.writeBytes(data);
            System.out.println("Data written to file successfully");
            out.close();
            dout.close();
            sc.close();
        }
        catch (FileNotFoundException e) {
            System.out.println("Files does not exist");
        }
        catch (IOException e) {
            System.out.println("Error in writing to a file");
        }
    }
}
```

Choose the correct option regarding the code.

Options :

6406531884690. ✓ This program prints the message:
Data written to file successfully

6406531884691. ✗ This program prints the message:
Files does not exist

6406531884692. ✗

This program prints the message:
Error in writing to a file

6406531884693. ✖ This program terminates abnormally due to unhandled exceptions.

Question Number : 88 Question Id : 640653563802 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 4

Question Label : Multiple Choice Question

Consider the Java code given below.

You may make use of the description given below.

The logarithm of a number n to the base x is the number of times x has to be multiplied with itself so get n , and is denoted by $\log_x(n)$.

For example, $\log_2(4) = 2$, $\log_{10}(100) = 2$. Note that $\log_x(1) = 1$.

```
class Logarithm{
    private int base;
    // Constructor to initialize the instance variable base
    public int log(int num) {
        assert num > 0; //LINE 1
        if (num == 1)
            return 0;
        assert base > 0; //LINE 2
        return 1 + log(num/base);
    }
}

public class AssertionTest {
    public static void main(String[] args) {
        Logarithm obj1 = new Logarithm(10);
        Logarithm obj2 = new Logarithm(-2);

        System.out.println(obj1.log(1000));
        System.out.println(obj2.log(-2));
    }
}
```

Choose the correct option when the program is executed as:

```
java -ea AssertionTest
```

Options :

This program generates the output:

3

1

6406531884694. ✖

This program throws AssertionError at LINE 1 after printing the value:

3

6406531884695. ✔

This program throws AssertionError at LINE 2 after printing the value:

3

6406531884696. ✖

This program terminates abnormally after printing the value:

3

6406531884697. ✖

Question Number : 89 Question Id : 640653563803 Question Type : MCQ Is Question

Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 4

Question Label : Multiple Choice Question

Consider the Java program given below.

```

import javax.swing.*;
public class GUITest extends JFrame{
    JPanel inputPanel,outputPanel,opPanel;
    JLabel label1,label2;
    JRadioButton sq,cube;
    JTextField input;
    JButton button;
    public GUITest() {
        label1 = new JLabel("Enter Number");
        input = new JTextField(10);
        button = new JButton("Submit");
        inputPanel = new JPanel();
        // label1, input and button are added to inputPanel

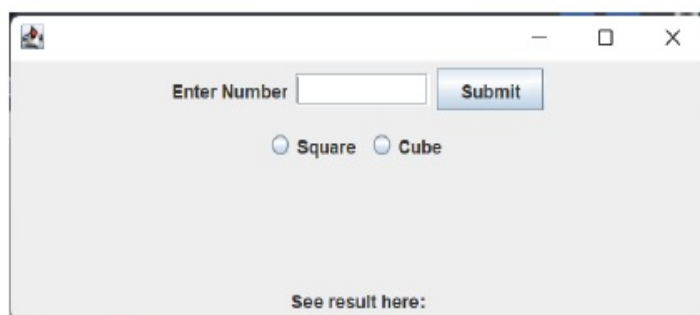
        sq = new JRadioButton("Square");
        cube = new JRadioButton("Cube");
        opPanel = new JPanel();
        // sq and cube are added to opPanel

        label2 = new JLabel("See result here:");
        outputPanel = new JPanel();
        // label2 is added to outputPanel

        setVisible(true);
        //CODE BLOCK
    }
    public static void main(String[] args) {
        new GUITest();
    }
}

```

Choose the correct option to be filled in place of CODE BLOCK such that the above program produces the GUI given below:



Options :

6406531884698. ✖

```

        add(inputPanel,"Center");
        add(opPanel,"North");
        add(outputPanel,"South");

```

6406531884699. ✖

```
add(inputPanel,"North");  
add(opPanel,"South");  
add(outputPanel,"Center");
```

```
add(inputPanel,"North");  
add(opPanel,"Center");  
6406531884700. ✓ add(outputPanel,"South");
```

```
add(inputPanel,"South");  
add(opPanel,"Center");  
6406531884701. ✗ add(outputPanel,"North");
```

Question Number : 90 Question Id : 640653563804 Question Type : MCQ Is Question

Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 4

Question Label : Multiple Choice Question

Consider the Java program given below.

```

class Counter implements Runnable{
    boolean stopRequested = false;
    long count = 0;
    public void run() {
        while (!stopRequested) {
            count++;
            if (count==1000000) {
                stopRequested = true;
            }
        }
    }
    public void setStop(boolean stop){
        stopRequested = stop;
    }
    public long getCount(){
        return count;
    }
}

public class ThreadEx {
    public static void main(String[] args) throws InterruptedException {
        Counter ctr = new Counter();
        Thread backgroundThread = new Thread(ctr);
        backgroundThread.start();
        Thread.sleep(1);
        ctr.setStop(true);
        System.out.println(ctr.getCount());
    }
}

```

What will the output be?

Options :

6406531884702. ✖ 0

6406531884703. ✖ 1000000

6406531884704. ✔ Some whole number between 0 and 1000000

6406531884705. ✖ 999999

Sub-Section Number :

3

Sub-Section Id :

64065380369

Question Shuffling Allowed :

Yes

Is Section Default? :

null

Question Number : 91 Question Id : 640653563793 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 5

Question Label : Multiple Choice Question

Consider the code given below.

```
interface Storable{
    void store();
}
interface Transferable{
    void transfer();
}
class Laptop{
    public HardDisk getDisk() {
        return new HardDisk();
    }
    private class HardDisk implements Storable,Transferable {
        public void store() {
            System.out.println("Stores data");
        }
        public void transfer() {
            System.out.println("Transfers data");
        }
    }
}
public class PrivateTest {
    public static void main(String[] args) {
        // CODE BLOCK
    }
}
```

Choose the correct option to be filled in place of CODE BLOCK so that the output is:

Stores data
Transfers data

Options :

```
Storable obj1 = new Laptop().getDisk();
Transferable obj2 = new Laptop().getDisk();
obj1.store();
obj2.transfer();
```

6406531884657. ✓


```
Storable obj1 = new Laptop().getDisk();  
obj1.store();  
obj1.transfer();
```

6406531884658. ✖

```
Transferable obj1 = new Laptop().getDisk();  
obj1.store();  
obj1.transfer();
```

6406531884659. ✖

```
HardDisk obj1 = new Laptop().getDisk();  
obj1.store();  
obj1.transfer();
```

6406531884660. ✖

Sub-Section Number :	4
Sub-Section Id :	64065380370
Question Shuffling Allowed :	Yes
Is Section Default? :	null

Question Number : 92 Question Id : 640653563784 Question Type : MSQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 5 Selectable Option : 0

Question Label : Multiple Select Question

Consider the code given below.

```

1  interface Readable{
2      void read();
3  }
4  interface Displayable{
5      void display();
6  }
7  interface Computable extends Displayable, Readable{
8      void compute();
9  }
10 abstract class Calculator implements Computable{
11     public void compute(){
12         System.out.println("Calculator computes");
13     }
14 }
15 class Phone extends Calculator{
16     public void read(){
17         System.out.println("Phone reads");
18     }
19     public void display(){
20         System.out.println("Phone display");
21     }
22 }
23 public class Test{
24     public static void main(String[] args){
25         Readable r = new Phone();
26         r.read();
27     }
28 }

```

Choose the correct option/s

Options :

6406531884620. ✖ Compilation error at LINE 7 because an interface cannot *extend* two interfaces.

6406531884621. ✖ Compilation error in LINE 10 to LINE 14 because the abstract class Calculator cannot implement an interface.

6406531884622. ✖ Compilation error at LINE 25 because a reference variable of interface Readable cannot refer to an object of class Phone.

6406531884623. ✔ The program generates the output: Phone reads

Question Number : 93 Question Id : 640653563787 Question Type : MSQ Is Question

Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 5 Selectable Option : 0

Question Label : Multiple Select Question

Consider the Java code given below that prints the seating capacity of each vehicle in a list of vehicles. From among the options, identify the appropriate function header for function printSeating that takes as input a list of Vehicle objects and prints the seating capacity of each.

```
import java.util.*;
abstract class Vehicle{
    abstract void capacity();
}
class Bike extends Vehicle{
    public void capacity() {
        System.out.println("Capacity is two");
    }
}
class Auto extends Vehicle{
    public void capacity() {
        System.out.println("Capacity is three");
    }
}
public class Test {
    // LINE 1: FUNCTION HEADER
    {
        // invokes method capacity()
        // to print the capacity of each vehicle
    }
    public static void main(String[] args) {
        List<Vehicle> vlist = new ArrayList<Vehicle>();
        vlist.add(new Bike());
        vlist.add(new Auto());
        seating(vlist);
    }
}
```

Choose the correct option(s).

Options :

6406531884632. ✖ `public static void seating(List<Bike> obj)`

6406531884633. ✔ `public static <T extends Vehicle> void seating(List<T> obj)`

6406531884634. ✖ `public static void seating(List<Auto> obj)`

6406531884635. ✔ `public static void seating(List<? extends Vehicle> obj)`

Question Number : 94 Question Id : 640653563788 Question Type : MSQ Is Question

Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 5 Selectable Option : 0

Question Label : Multiple Select Question

Which of the following statements counts the number of integers between 100 and 150 (including 100 and 150) that are not divisible by 2?

Options :

6406531884636. ✔ `Stream.iterate(100, n -> n + 1).limit(51).filter(n -> n % 2 != 0).count();`

6406531884637. ✖ `Stream.iterate(100, n -> n + 1).filter(n -> n % 2 != 0).count();`

6406531884638. ✔ `Stream.iterate(100, n-> n <= 150, n -> n + 1).filter(n -> n % 2 != 0).count();`

6406531884639. ✖ `Stream.filter(100, n-> n < 150, n -> n + 1).iterate(n -> n % 2 == 0).count();`

Question Number : 95 Question Id : 640653563789 Question Type : MSQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 5 Selectable Option : 0

Question Label : Multiple Select Question

Consider the code given below.

```
import java.util.*;
class Account{
    int balance;
    public synchronized void withdraw(int amount) {
        if(balance > amount)
            balance = balance - amount;
        System.out.println("After withdrawing, balance = "+balance);
    }
    public synchronized void deposit(int amount) {
        balance = balance + amount;
        System.out.println("After depositing, balance = "+balance);
    }
}
class User extends Thread{
    Account acc;
    public User(Account obj){
        acc = obj;
    }
    public void run() {
        acc.withdraw(500);
        acc.deposit(1000);
    }
}
public class Test{
    public static void main(String args[]) {
        Account my_acc = new Account();
        User u1 = new User(my_acc);
        User u2 = new User(my_acc);
        u1.start();
        u2.start();
    }
}
```

Choose all the options that would NEVER occur as a result of execution of this code.

Options :

After withdrawing, balance = 0
After depositing, balance = 1000
After withdrawing, balance = 500
6406531884640. ✖ After depositing, balance = 1500

After withdrawing, balance = 0
After withdrawing, balance = 0
After depositing, balance = 1000
6406531884641. ✖ After depositing, balance = 2000

After withdrawing, balance = 0
After depositing, balance = 1000
After depositing, balance = 2000
6406531884642. ✔ After withdrawing, balance = 1000

After depositing, balance = 1000
After withdrawing, balance = 500
After withdrawing, balance = 1000
6406531884643. ✔ After depositing, balance = 2000

Question Number : 96 Question Id : 640653563791 Question Type : MSQ Is Question

Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 5 Selectable Option : 0

Question Label : Multiple Select Question

Consider the code given below.


```

import java.util.*;
public class Person{
    private String name;
    private int age;
    public Person(String n, int a) {
        name = n;
        age = a;
    }
    public int getAge(){
        return age;
    }
    public void print() {
        System.out.println(name + " : " + age);
    }
}

public class FClass{
    public static void main(String[] args) {
        var list = new ArrayList<Person>();
        list.add(new Person("Robin", 33));
        list.add(new Person("Indra", 76));
        list.add(new Person("Smita", 35));
        list.add(new Person("Rikki", 26));
        Collections.sort(list, _____); LINE 1
        for(var l: list)
            l.print();
    }
}

```

Identify the appropriate option(s) to fill in the blank at LINE 1, such that the output is:

```

Indra : 76
Smita : 35
Robin : 33
Rikki : 26

```

Options :

6406531884649. ✖ (a, b) -> a.getAge() - b.getAge()

6406531884650. ✔ (a, b) -> b.getAge() - a.getAge()

6406531884651. ✔ (Person a, Person b) -> a.getAge() - b.getAge()

6406531884652. ✖ (a, b) -> { return b.getAge() - a.getAge(); }

Question Number : 97 Question Id : 640653563794 Question Type : MSQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 5 Selectable Option : 0

Question Label : Multiple Select Question

Consider the code given below.

```
import java.util.*;
public class ListExample {
    public static void main(String[] args) {
        ----- //LINE 1
        ----- //LINE 2
        list1 = new ArrayList<String>();
        list1.add("India");
        list1.add("Hyderabad");
        list2 = new LinkedList<String>(list1);
        for(String s : list1)
            System.out.println(s);
        for(String s : list2)
            System.out.println(s);
    }
}
```

If the code given above produces the output:

India
Hyderabad
India
Hyderabad

What should be the correct choice for LINE 1 and LINE 2?

Options :

6406531884661. ✔ List<String> list1;
List<String> list2;

6406531884662. ✖

```
AbstractSequentialList<String> list1;  
AbstractSequentialList<String> list2;
```

6406531884663. ✖

```
AbstractList<String> list1;  
SequentialList<String> list2;
```

6406531884664. ✔

```
AbstractList<String> list1;  
AbstractSequentialList<String> list2;
```

Question Number : 98 Question Id : 640653563795 Question Type : MSQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 5 Selectable Option : 0

Question Label : Multiple Select Question

Consider the Java code given below.

```

import java.util.*;
public class SetTest {
    public static void main(String[] args) {
        ArrayList<String> list = new ArrayList<String>();
        list.add("Java");
        list.add("Programming");
        list.add("Python");
        list.add("Script");
        // CODE BLOCK
        for(String str:list) {
            set1.add(str);
            set2.add(str);
        }
        for(String str:set1)
            System.out.print(str+" ");
        System.out.println();
        for(String str:set2)
            System.out.print(str+" ");
    }
}

```

Choose the correct option(s) to be filled in place of CODE BLOCK so that the program always generates the following output:

```

Java Programming Python Script
Java Programming Python Script

```

Options :

6406531884665. ✖

```

        var set1 = new HashSet<String>();
        var set2 = new TreeSet<String>();

```

6406531884666. ✔

```

        var set1 = new LinkedHashSet<String>();
        var set2 = new TreeSet<String>();

```

6406531884667. ✔

```

        var set1 = new TreeSet<String>();
        var set2 = new TreeSet<String>();

```

6406531884668. ✖

```

        var set1 = new LinkedHashSet<String>();
        var set2 = new HashSet<String>();

```

MLT

Section Id :	64065338322
Section Number :	5
Section type :	Online
Mandatory or Optional :	Mandatory
Number of Questions :	21
Number of Questions to be attempted :	21
Section Marks :	100
Display Number Panel :	Yes
Group All Questions :	No
Enable Mark as Answered Mark for Review and Clear Response :	Yes
Maximum Instruction Time :	0
Sub-Section Number :	1
Sub-Section Id :	64065380371
Question Shuffling Allowed :	No
Is Section Default? :	null

Question Number : 99 Question Id : 640653563805 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 0

Question Label : Multiple Choice Question

THIS IS QUESTION PAPER FOR THE SUBJECT "DIPLOMA LEVEL : MACHINE LEARNING TECHNIQUES (COMPUTER BASED EXAM)"

ARE YOU SURE YOU HAVE TO WRITE EXAM FOR THIS SUBJECT?

CROSS CHECK YOUR HALL TICKET TO CONFIRM THE SUBJECTS TO BE WRITTEN.

(IF IT IS NOT THE CORRECT SUBJECT, PLS CHECK THE SECTION AT THE TOP FOR THE SUBJECTS