

Options :

6406532330067. ✓ $0.4x_1 + 0.6x_2$

6406532330068. ✓ $0.4x_1 + 0.2x_2 + 0.2x_3 + 0.2x_4$

6406532330069. ✖ $0.4x_1 - 0.2x_2 + 0.5x_3 + 0.3x_4$

6406532330070. ✖ $0.4x_1 + 0.2x_2 + 0.3x_3 + 0.3x_4$

Java

Section Id :	64065349264
Section Number :	5
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
Section Negative Marks :	0
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 :	640653103274
Question Shuffling Allowed :	No
Is Section Default? :	null

Question Number : 98 Question Id : 640653697675 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 :

6406532330071.  YES

6406532330072.  NO

Sub-Section Number : 2

Sub-Section Id : 640653103275

Question Shuffling Allowed : Yes

Is Section Default? : null

Question Number : 99 Question Id : 640653697676 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 below.

```

class Student {
    String name;
    public Student(String n) {
        name = n;
    }
    public String toString() {
        return "Name = " + name;
    }
}
class Leader extends Student {
    private int marks;
    public Leader(String n, int m) {
        super(n);
        marks = m;
    }
    public Leader(Leader l) {
        super(l.name);
        marks = l.marks;
    }
    public String toString() {
        return super.toString() + ", " + "Marks = " + marks;
    }
}
public class College {
    public static void main(String args[]) {
        Student l1 = new Leader("Rohan", 97);
        Student l2 = new Leader((Leader) l1);
        l2.name = "Ananya";
        System.out.println(l1 + "\n" + l2);
    }
}

```

What will the output be?

Options :

Name = Rohan, Marks = 97

6406532330073. ✖ Name = Rohan, Marks = 97

Name = Rohan, Marks = 97

6406532330074. ✔ Name = Ananya, Marks = 97

Name = Ananya, Marks = 97

6406532330075. ✖ Name = Ananya, Marks = 97

Name = Ananya, Marks = 97
6406532330076. ✖ Name = Rohan, Marks = 97

Question Number : 100 Question Id : 640653697677 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

Method `Optional.ofNullable(T value)` returns an `Optional` that describes the specific value, if non-null; otherwise returns an empty `Optional`.

Based on this description, consider the code given below, and answer the question that follows.

```
import java.util.*;
class GoldQC {
    String type;
    int karats;
    public GoldQC(int k){
        karats = k;
    }
    public GoldQC(String t, int k){
        type = t;
        karats = k;
    }
}
public class Test {
    public static void main(String[] args) {
        var goldList = new ArrayList<GoldQC>();
        goldList.add(new GoldQC("Ring", 24));
        goldList.add(new GoldQC(18));
        for (GoldQC g: goldList) {
            System.out.print("karats = " + g.karats + ", type = ");
            Optional<String> karatsValue = Optional.ofNullable(g.type);
            karatsValue.ifPresentOrElse(t -> System.out.println(t),
                                     () -> System.out.println("unknown")
            );
        }
    }
}
```

Choose the correct option.

Options :

6406532330077. ✖ This program generates the output:
karats = 24, type = null
karats = 18, type = unknown

6406532330078. ✖ This program generates the output:
karats = 24, type = Ring
karats = 18, type = null

6406532330079. ✔ This program generates the output:
karats = 24, type = Ring
karats = 18, type = unknown

6406532330080. ✖ This program terminates abnormally due to NullPointerException after printing:
karats = 24, type = Ring

Question Number : 101 Question Id : 640653697680 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 give below.

```
abstract class Operation implements Runnable {
    static int result = 0;
}
class AddOperation extends Operation {
    public void run() {
        if (result != 8) {
            result = result + 3;
        }
    }
}
class MultiplyOperation extends Operation {
    public void run() {
        if (result != 9) {
            result = result * 2;
        }
    }
}
public class MathOperations {
    public static void main(String[] args) {
        Operation addOp = new AddOperation();
        Operation multiplyOp = new MultiplyOperation();
        Thread t1 = new Thread(addOp);
        Thread t2 = new Thread(multiplyOp);
        t1.start();
        t2.start();
        System.out.println(Operation.result);
    }
}
```

Choose the correct option.

Options :

6406532330089. ✖ The program will never generate the output: 0

6406532330090. ✖ The program will never generate the output: 3

6406532330091. ✖ The program will never generate the output: 6

6406532330092. ✔ The output can be 0 or 3 or 6 .

Question Number : 102 Question Id : 640653697681 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.util.*;
public class Test {
    public static void main(String args[]) {
        Map<String, Integer> tmap, lmap, hmap;
        hmap = new HashMap<String, Integer>();
        hmap.put("Bhuvana", 6);
        hmap.put("Ann", 9);
        hmap.put("Greeshma", 12);
        hmap.put("Karan", 20);
        lmap = new LinkedHashMap<String, Integer>();
        tmap = new TreeMap<String, Integer>();
        for(Map.Entry<String, Integer> entry:hmap.entrySet()) {
            tmap.put(entry.getKey(), entry.getValue());
        }
        for(Map.Entry<String, Integer> entry:tmap.entrySet()) {
            lmap.put(entry.getKey(), entry.getValue());
        }
        System.out.println(hmap);
        System.out.println(tmap);
        System.out.println(lmap);
    }
}
```

Choose the correct option regarding predicting the order of elements, with respect to the original ordering, in each map object.

Options :

6406532330093. ✖ We cannot predict the order of elements in hmap, tmap or lmap.

6406532330094. ✖ We can predict the order of elements in hmap and lmap, but not tmap.

6406532330095. ✔ We can predict the order of elements in tmap and lmap but not that of hmap.

6406532330096. ✖ We can predict the order of elements in hmap, tmap and lmap.

Question Number : 103 Question Id : 640653697682 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.util.*;
import java.util.stream.*;
public class Test {
    public static void main(String[] args) {
        var list = new ArrayList<String>();
        list.add(null);
        list.add("Java");
        list.add(null);
        list.add("Python");
        list.add("Kotlin");
        list.add("CPP");
        list.add("CPP");
        list.add("Kotlin");
        Stream<String> stream = list.stream();
        // CODE BLOCK
        System.out.println(obj);
    }
}
```

Choose the correct option(s) to fill in place of CODE BLOCK so that the output is:
[CPP, Java, Kotlin, Python]

Options :

```
var obj = list.stream()
    .filter(str -> str != null)
    .limit(4)
    .collect(Collectors.toSet());
```

6406532330097. ✖

```
var obj = list.stream()
    .map(str -> str != null)
    .collect(Collectors.toList());
```

6406532330098. ✖

```
var obj= list.stream()
                .filter(str -> str != null)
                .collect(Collectors.toCollection(TreeSet::new));
```

6406532330099. ✓

```
var obj = list.stream()
                .filter(str -> str != null)
                .limit(4)
                .collect(Collectors.toList());
```

6406532330100. ✖

Question Number : 104 Question Id : 640653697684 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.io.*;
class Product implements Serializable {
    private String name = "****";
    private double price = 0.0;
    private transient String discountCode = "****";
    public Product(String name, double price, String discountCode) {
        this.name = name;
        this.price = price;
        this.discountCode = discountCode;
    }
    public String toString() {
        return "Product_name = " + name + ", Price = " + price + ", Discount Code
        = " + discountCode;
    }
}

public class Test {
    public static void main(String[] args) throws Exception {
        var fos = new FileOutputStream("product.txt");
        var oos = new ObjectOutputStream(fos);
        oos.writeObject(new Product("Laptop", 1200.00, "DISCOUNT123"));
        oos.close();
        var fis = new FileInputStream("product.txt");
        var ois = new ObjectInputStream(fis);
        Product p = (Product) ois.readObject();
        ois.close();
        System.out.println(p);
    }
}
```

What will the output be?

Options :

6406532330105. ✖ Product_name = Laptop, Price = 1200.0, Discount Code = DISCOUNT123

6406532330106. ✖ Product_name = Laptop, Price = null, Discount Code = null

6406532330107. ✔ Product_name = Laptop, Price = 1200.0, Discount Code = null

6406532330108. ✖ Product_name = Laptop, Price = 1200.0, Discount Code = ****

Question Number : 105 Question Id : 640653697686 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

Method `Collectors.summarizingInt(mapping function)` returns a `Collector` which applies an int-producing mapping function to each input element, and returns summary statistics, such as count, min, max, sum, and average, for the resulting values.

```
import java.util.*;
import java.util.stream.*;
public class Test {
    public static void main(String[] args) {
        var salaryList = new ArrayList<Integer>();
        salaryList.add(50000);
        salaryList.add(60000);
        salaryList.add(75000);
        salaryList.add(90000);
        salaryList.add(55000);
        IntSummaryStatistics stat = salaryList.stream()
                                                .collect(Collectors.summarizingInt(x -> x));
        System.out.println(stat.getMin());
        System.out.println(stat.getMax());
        System.out.println(stat.getSum());
        System.out.println(stat.getAverage());
    }
}
```

Choose the correct option.

Options :

6406532330113. ✖ This program gives compile time error because a list cannot be converted to a stream.

6406532330114. ✖ This program generates compile time error with the message: incompatible types: possible lossy conversion from double to int

6406532330115. ✖ This program generates compile time error with the message: Cannot invoke `stream()` on the primitive type `int`

This program generates the output:

50000

90000

330000

6406532330116. ✓ 66000.0

Question Number : 106 Question Id : 640653697689 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.

```
public class Test {  
    public static int factorial(int n) {  
        int result = 1;  
        assert n >= 0 : "n must be >= 0";    // assert-1  
        for (int i = 1; i <= n; i++) {  
            result *= i;  
        }  
        assert result >= 0 : result;          // assert-2  
        return result;  
    }  
    public static void main(String[] args) {  
        int number = -7;  
        assert number != 0 : "number == 0"; // assert-3  
        int fact = factorial(number);  
        assert fact >= 0 : fact;             // assert-4  
    }  
}
```

Identify the first assert statement that throws the AssertionError when the class is executed as: `java -ea Test`

Options :

6406532330125. ✓ assert-1

6406532330126. ✗ assert-2

6406532330127. ✖ assert-3

6406532330128. ✖ assert-4

Question Number : 107 Question Id : 640653697690 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 NegativeValueException extends Exception {
    public String toString() {
        return "Negative value encountered during update";
    }
}
public class Test {
    public static void update(int[] array, int index) throws NegativeValueException{
        if (array[index] < 0) {
            throw new NegativeValueException();
        }
        array[index] = array[index] * 3;
    }
    public static void main(String[] args) {
        int[] arr = {2, 8, -5, 10, 12};
        try {
            for (int i = 0; i < arr.length; i++) {
                update(arr, i);
            }
        } catch (NegativeValueException e) {
            System.out.println(e);
        }
        for (int n : arr) {
            System.out.print(n + " ");
        }
    }
}
```

What will the output be?

Options :

6406532330129. ✖ Negative value encountered during update

6406532330130. ✖ 6 24 -5 10 12

Negative value encountered during update
6406532330131. ✖ 6 24

Negative value encountered during update
6406532330132. ✔ 6 24 -5 10 12

Question Number : 108 Question Id : 640653697692 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 Test{
    public static void main(String[] args) {
        PriorityQueue<String> queue1 = new PriorityQueue<String>();
        queue1.add("LG");
        queue1.add("Samsung");
        queue1.add("Panasonic");
        queue1.add("Bosch");
        queue1.add("Whirlpool");
        ArrayDeque<String> queue2 = new ArrayDeque<String>();
        while(queue1.size()>0) {
            queue2.addLast(queue1.poll()); //LINE-1
        }
        System.out.print(queue2);
    }
}
```

Choose the correct option.

Options :

This program generates the output:

6406532330137. ✓ [Bosch, LG, Panasonic, Samsung, Whirlpool]

This program generates the output:

6406532330138. ✗ [LG, Samsung, Panasonic, Bosch, Whirlpool]

6406532330139. ✗ [Whirlpool, Bosch, Panasonic, Samsung, LG]

6406532330140. ✗ Compilation error at LINE-1

Question Number : 109 Question Id : 640653697693 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 Vehicle {
    public void start() {
        System.out.println("Vehicle started");
    }
    public void drive() {
        System.out.println("Driving");
    }
}
class Car extends Vehicle {
    public void drive() {
        System.out.println("Driving a car");
    }
    public void park() {
        System.out.println("Parked the car");
    }
}
class ElectricCar extends Car {
    public void charge() {
        System.out.println("Charging the electric car");
    }
}
public class Test {
    public static void main(String[] args) {
        Vehicle v = new ElectricCar();
        v.start();
        v.drive();
        v.charge();    //LINE 1
    }
}
```

Choose the correct option.

Options :

This code generates the output:

Vehicle started

Driving a car

Charging the electric car

This code generates the output:

Vehicle started

Driving

6406532330142. ✖ Parked the car

6406532330143. ✔ LINE 1 generates compilation error.

This code generates the output:

Vehicle started

Driving a car

6406532330144. ✖ Parked the car

Question Number : 110 Question Id : 640653697694 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 Marks{
    public static void main(String[] args){
        Map<String, Integer> quiz1 = new TreeMap<String, Integer>();
        quiz1.put("Sita", 300);
        quiz1.put("Geeta", 200);
        quiz1.put("Latha", 300);
        quiz1.put("Reena", 340);

        Map<String, Integer> quiz2 = new TreeMap<String, Integer>();
        quiz2.put("Sita", 400);
        quiz2.put("Latha", 400);
        quiz2.put("Reena", 360);
        quiz2.put("Pragna", 400);

        Map<String, Integer> avg = new TreeMap<String, Integer>();

        for(Map.Entry<String, Integer> m1 : quiz1.entrySet())
            avg.put(m1.getKey(), m1.getValue());

        for(Map.Entry<String, Integer> m2 : quiz2.entrySet())
            avg.merge(m2.getKey(), m2.getValue(), (x, y) -> (x + y)/2); // LINE 1

        System.out.println(avg);
    }
}
```

Choose the correct option.

Options :

6406532330145. ✖ Compile time error at LINE 1 because of invalid key

This program generates the output:

6406532330146. ✔ {Geeta=200, Latha=350, Pragna=400, Reena=350, Sita=350}

This program generates the output:

6406532330147. ✖ {Sita=350, Geeta=200, Latha=350, Reena=350, Pragna=400}

This program generates the output:

6406532330148. ✖ {Latha=350, Reena=350, Sita=350}

Question Number : 111 Question Id : 640653697696 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 Example {  
2      public void methodOne() {  
3          // ...  
4          methodTwo();  
5          // ...  
6      }  
7      public void methodTwo() {  
8          // ...  
9      }  
10     public static void methodThree() {  
11         // ...  
12         Example example = new Example();  
13         example.methodOne();  
14         // ...  
15     }  
16     public static void main(String[] args) {  
17         // ...  
18         methodThree();  
19     }  
20 }
```

During the execution of Line 14 in the given code, which method's activation record is at the top of the stack of activation records?

Options :

6406532330153. ✖ main

6406532330154. ✖ methodOne

6406532330155. ✖ methodTwo

6406532330156. ✓ methodThree

Sub-Section Number : 3
Sub-Section Id : 640653103276
Question Shuffling Allowed : Yes
Is Section Default? : null

Question Number : 112 Question Id : 640653697688 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.

```
class Student implements Cloneable{
    String name;
    int[] marks;
    public Student(String n, int[] m) {
        name = n;
        marks = m;
    }
    public Object clone() throws CloneNotSupportedException{
        return super.clone();
    }
}

public class Cloning {
    public static void main(String[] args) throws CloneNotSupportedException{
        int[] m = {12, 13, 15};
        Student s1 = new Student("Ram", m);
        Student s2 = (Student)s1.clone();
        Student s3 = s1;
        s2.marks[1] = 34;
        s3.name = "Sita";
        System.out.println(s1.name + " " + s1.marks[1]);
        System.out.println(s2.name + " " + s2.marks[1]);
        System.out.println(s3.name + " " + s3.marks[1]);
    }
}
```

What will the output be?

Options :

Sita 34
Ram 34
6406532330121. ✓ Sita 34

Sita 13
Ram 34
6406532330122. ✗ Sita 13

Ram 13
Ram 34
6406532330123. ✗ Sita 13

Ram 34
Ram 34
6406532330124. ✗ Sita 34

Question Number : 113 Question Id : 640653697691 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 two Java files given below.

Test1.java:

```
package pack1;
public class Test1 {
    protected void f1() {
        System.out.println("Inside Test1");
    }
}
```

Test2.java:

```
package pack2;
import pack1.Test1; //LINE-1
public class Test2 extends Test1{ //LINE-2
    public static void main(String[] args) {
        Test2 obj = new Test2();
        obj.f2();
    }
    public void f2() {
        f1(); //LINE-3
    }
}
```

Choose the correct option.

Options :

The program generates the output:

6406532330133. ✓ Inside Test1

6406532330134. ✗ Runtime error, because custom packages cannot be imported (see LINE-1).

6406532330135. ✗ LINE-2 generates a compilation error because of extending class Test1, which is located in another package.

6406532330136. ✗ LINE-3 generates a compilation error because method f1() is not visible from class Test2.

Sub-Section Number :

4

Sub-Section Id :

640653103277

Question Shuffling Allowed :

Yes

Is Section Default? :

null

Question Number : 114 Question Id : 640653697683 Question Type : MCQ Is Question

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

Correct Marks : 6

Question Label : Multiple Choice Question

ConcurrentHashMap is a hash table implementation that supports full concurrency of retrievals and high expected concurrency for updates. All operations on this collection are thread-safe. Consider the code given below that uses a ConcurrentHashMap, and answer the question that follows.

```
class Example extends Thread{
    Map siMap;
    //Constructor is defined here
    public void run(){
        siMap.put("D",4);
    }
}

public class FClass{
    public static void main (String[] args) {
        Map<String, Integer> siMap = new ConcurrentHashMap<String, Integer>();
        String[] str = {"A", "B", "C"};
        Integer[] arr = {1, 2, 3};
        for(int i = 0; i < str.length; i++){
            siMap.put(str[i],arr[i]);
        }
        Example t = new Example(siMap);
        t.start();
        Set s = siMap.entrySet();
        Iterator itr = s.iterator();
        while(itr.hasNext()){
            Map.Entry m = (Map.Entry)itr.next();
            System.out.println(m.getKey() + " => " + m.getValue());
        }
    }
}
```

Which of the following is NOT true about the given code?

Options :

6406532330101. ✓ This program may generate ConcurrentModificationException.

This program generates the output:

A => 1

B => 2

C => 3

6406532330102. ✖ D => 4

This program generates the output:

D => 4

A => 1

B => 2

6406532330103. ✖ C => 3

This program generates the output:

A => 1

B => 2

6406532330104. ✖ C => 3

Question Number : 115 Question Id : 640653697698 Question Type : MCQ Is Question

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

Correct Marks : 6

Question Label : Multiple Choice Question

FileOutputStream(String name,boolean append) method creates a file output stream to write to the file with the specified name. If the second argument is true, then bytes will be written to the end of the file rather than the beginning.

Consider the code given below. Assume that, before execution of the given code, the files "file1.txt" and "file2.txt" have the following text in them.

Hello friend

```
public class Test {
    public static void main(String[] args) {
        try {
            var out = new FileOutputStream("file1.txt", true);
            var dout = new DataOutputStream(out);
            dout.writeBytes(", What's up?");
            dout.close();
            var out2 = new FileOutputStream("file2.txt", false);
            var dout2 = new DataOutputStream(out2);
            dout2.writeBytes(", What's up?");
            dout2.close();
        }
        catch(IOException e) {
            System.out.println(e);
        }
    }
}
```

Choose the correct option regarding the contents of file1.txt and file2.txt after the program finishes execution.

Options :

file1.txt:

, What's up?

file2.txt:

6406532330161. ✖ Hello friend, What's up?

file1.txt and file2.txt

6406532330162. ✖ Hello friend, What's up?

file1.txt:

Hello friend, What's up?

file2.txt:

6406532330163. ✔ , What's up?

file1.txt and file2.txt

6406532330164. ✖ , What's up?

Sub-Section Number :	5
Sub-Section Id :	640653103278
Question Shuffling Allowed :	Yes
Is Section Default? :	null

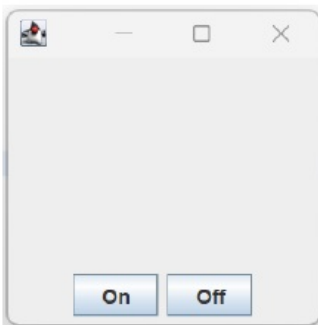
Question Number : 116 Question Id : 640653697687 Question Type : MSQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 6 Max. Selectable Options : 0

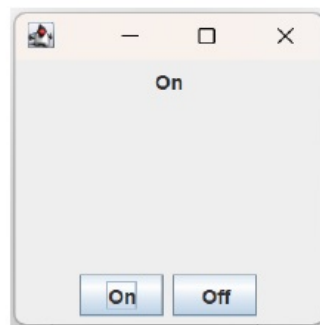
Question Label : Multiple Select Question

Consider the code given below.

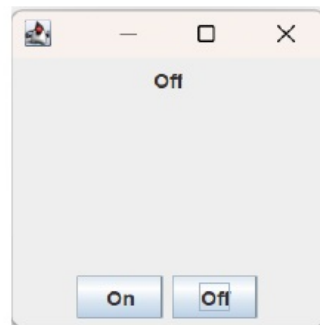
```
import javax.swing.*;
import java.awt.event.*;
public class ButtonEvents extends JFrame implements ActionListener {
    private JButton b1, b2;
    private JLabel l1;
    JPanel pnl1, pnl2;
    public ButtonEvents() {
        b1 = new JButton("On");
        b2 = new JButton("Off");
        pnl1 = new JPanel();
        //add b1 and b2 to pnl1
        add(pnl1, "South");
        l1 = new JLabel("");
        pnl2 = new JPanel();
        //add l1 to pnl2
        add(pnl2, "North");
        setVisible(true);
        setSize(200, 200);
        b1.setActionCommand("s1");
        b2.setActionCommand("s2");
        b1.addActionListener(this);
        b2.addActionListener(this);
    }
    public void actionPerformed(ActionEvent e) {
        // CODE BLOCK
    }
    public static void main(String[] args) {
        new ButtonEvents();
    }
}
```



(a) Initial appearance



(b) Clicking button "On"



(c) Clicking button "Off"

Figure 1: GUI

Choose the correct code segment inside method `actionPerformed()` such that whenever either of the two buttons (On/Off) is clicked, the label text of `pnl2` will change accordingly as shown in Figure 1.

Options :

```
if(e.getSource().equals(b1))
    l1.setText("On");
else if(e.getSource().equals(b2))
    l1.setText("Off");
```

6406532330117. ✓

6406532330118. ✗

```
if(e.getActionCommand().equals(b1))
    l1.setText("On");
else if(e.getActionCommand().equals(b2))
    l1.setText("Off");
```

```
        if(e.getActionCommand().equals("s1"))
            l1.setText("On");
        else if(e.getActionCommand().equals("s2"))
            l1.setText("Off");
```

6406532330119. ✓

```
        if(e.getSource().equals("s1"))
            l1.setText("On");
        else if(e.getSource().equals("s2"))
            l1.setText("Off");
```

6406532330120. ✗

Sub-Section Number :	6
Sub-Section Id :	640653103279
Question Shuffling Allowed :	Yes
Is Section Default? :	null

Question Number : 117 Question Id : 640653697678 Question Type : MSQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 4 Max. Selectable Options : 0

Question Label : Multiple Select Question

Which of the following statements count(s) the number of integers between 200 and 300 (including 200 and 300) that are not divisible by 7?

Options :

```
Stream.iterate(200, n -> n + 1).limit(101).filter(n -> n % 7 != 0)
6406532330081. ✓ .count();
```

```
Stream.iterate(200, n -> n + 1).filter(n -> n % 7 != 0)
6406532330082. ✗ .count();
```

```
Stream.iterate(200, n-> n <= 300, n -> n + 1).filter(n -> n % 7 != 0)
6406532330083. ✓ .count();
```

```
Stream.filter(200, n-> n <= 300, n -> n + 1).iterate(n -> n % 7 == 0)
6406532330084. ✖ .count();
```

Question Number : 118 Question Id : 640653697679 Question Type : MSQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 4 Max. Selectable Options : 0

Question Label : Multiple Select Question

Consider the code given below.

```
import java.util.*;
class GiftBasket {
    private int giftsCount;
    public synchronized void removeGifts(int count) {
        if (giftsCount > count)
            giftsCount = giftsCount - count;
        System.out.println("After removing gifts, remaining count = " + giftsCount);
    }
    public synchronized void addGifts(int count) {
        giftsCount = giftsCount + count;
        System.out.println("After adding gifts, total count = " + giftsCount);
    }
}
class Child extends Thread {
    GiftBasket giftBasket;
    public Child(GiftBasket g) {
        this.giftBasket = g;
    }
    public void run() {
        giftBasket.removeGifts(2);
        giftBasket.addGifts(5);
    }
}
public class Test {
    public static void main(String[] args) {
        GiftBasket giftBasket = new GiftBasket();
        Child c1 = new Child(giftBasket);
        Child c2 = new Child(giftBasket);
        c1.start();
        c2.start();
    }
}
```

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

Options :

After removing gifts, remaining count = 0

After adding gifts, total count = 5

After removing gifts, remaining count = 3

6406532330085. ✖ After adding gifts, total count = 8

6406532330086. ✖

After removing gifts, remaining count = 0
After removing gifts, remaining count = 0
After adding gifts, total count = 5
After adding gifts, total count = 10

After removing gifts, remaining count = 0
After adding gifts, total count = 5
After adding gifts, total count = 10
6406532330087. ✓ After removing gifts, remaining count = 5

After adding gifts, total count = 5
After removing gifts, remaining count = 3
After removing gifts, remaining count = 5
6406532330088. ✓ After adding gifts, total count = 10

Question Number : 119 Question Id : 640653697685 Question Type : MSQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0
Correct Marks : 4 Max. Selectable Options : 0
Question Label : Multiple Select Question

Consider the code given below.

```
interface Brewable {
    void brew();
}
interface Drinkable extends Brewable {
    void drink();
}
class CoffeeMachine {
    public Drinkable getEspressoMaker() {
        return new EspressoMaker();
    }
    private class EspressoMaker implements Drinkable {
        public void brew() {
            System.out.println("Brewing espresso in a coffee machine");
        }
        public void drink() {
            System.out.println("Enjoying a cup of espresso");
        }
    }
}
public class Test {
    public static void main(String[] args) {
        CoffeeMachine cm = new CoffeeMachine();
        // CODE BLOCK
    }
}
```

Choose the correct option(s) to fill in place of CODE BLOCK so that the output is:
Brewing espresso in a coffee machine
Enjoying a cup of espresso

Options :

```
Brewable b = cm.getEspressoMaker();
b.brew();
Drinkable d = b;
6406532330109. ✖ d.drink();
```

```
Drinkable d = cm.getEspressoMaker();
d.brew();
6406532330110. ✔ d.drink();
```

```
Brewable b = cm.getEspressoMaker();
b.brew();
6406532330111. ✖ b.drink();
```

```
Brewable b = cm.getEspressoMaker();  
b.brew();  
6406532330112. ✓ ((Drinkable)b).drink();
```

Question Number : 120 Question Id : 640653697695 Question Type : MSQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 4 Max. Selectable Options : 0

Question Label : Multiple Select Question

Consider the code given below.

```
abstract class Vehicle {  
    abstract void start(); // LINE-1  
    public void drive() { // LINE-2  
        System.out.println("Driving vehicle");  
    }  
}  
class Car extends Vehicle {  
    void start() {  
        System.out.println("Car started");  
    }  
}  
class Bike extends Vehicle {  
    void start() {  
        System.out.println("Bike started");  
    }  
}  
public class TestAbstract {  
    public static void main(String[] args) {  
        Vehicle[] v = new Vehicle[2];  
        v[0] = new Car(); // LINE-3  
        v[1] = new Bike(); // LINE-4  
        for (int i = 0; i < v.length; i++) {  
            v[i].start();  
            v[i].drive();  
        }  
    }  
}
```

Choose the correct option(s) regarding the above code.

Options :

6406532330149. ✖ Compilation error at LINE 1 because an abstract method should be public in an abstract class.

6406532330150. ✖ Compilation error at LINE 2 because you should write only abstract methods in an abstract class.

6406532330151. ✖ Compilation errors at LINE 3 and LINE 4 because it is illegal to add objects of type Car and Bike to Vehicle array.

6406532330152. ✔ This program generates the output:
Car started
Driving vehicle
Bike started
Driving vehicle

Question Number : 121 Question Id : 640653697697 Question Type : MSQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0
Correct Marks : 4 Max. Selectable Options : 0
Question Label : Multiple Select Question

Consider the Java code given below that prints the vehicle with highest mileage. From among the options, identify the appropriate function header for the function `printHighestMileage` that takes as input an array of `Vehicle` objects and prints the vehicle with highest mileage among them.

```
import java.util.*;
abstract class Vehicle {
    public abstract double getMileage();
}
class Car extends Vehicle {
    // getMileage() method that returns the mileage of the car
}
class Motorcycle extends Vehicle {
    // getMileage() method that returns the mileage of the motorcycle
}
public class Test {
    // LINE 1 : FUNCTION HEADER
    {
        //invokes method getMileage()
        //to print the value of highest mileage
    }
    public static void main(String[] args) {
        Vehicle[] vehicles = { new Car(), new Motorcycle() };
        printHighestMileage(vehicles);
    }
}
```

Choose the correct option(s).

Options :

6406532330157. ✓ `public static void printHighestMileage(Vehicle[] v)`

6406532330158. ✗ `public static void printHighestMileage(T[] v)`

6406532330159. ✓ `public static <T extends Vehicle> void printHighestMileage(T[] v)`

6406532330160. ✗ `public static void printHighestMileage(<?> v)`

AppDev2

Section Id :	64065349265
Section Number :	6
Section type :	Online
Mandatory or Optional :	Mandatory
Number of Questions :	33
Number of Questions to be attempted :	33
Section Marks :	100
Display Number Panel :	Yes
Section Negative Marks :	0
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 :	640653103280
Question Shuffling Allowed :	No
Is Section Default? :	null

Question Number : 122 Question Id : 640653697699 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 : MODERN APPLICATION DEVELOPMENT II (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