$$\text{span}\left\{\begin{pmatrix}1\\2\\0\end{pmatrix}, \begin{pmatrix}2\\3\\1\end{pmatrix}\right\}$$

Java

Section Id :	64065364124
Section Number :	11
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 :	No
Maximum Instruction Time :	0
Sub-Section Number :	1
Sub-Section Id :	640653134045
Question Shuffling Allowed :	No

Question Number : 302 Question Id : 640653903873 Question Type : MCQ Calculator : Yes

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 :

6406533043675. ✔ YES

6406533043676. ✖ NO

Sub-Section Number :	2
Sub-Section Id :	640653134046
Question Shuffling Allowed :	Yes

Question Number : 303 Question Id : 640653903874 Question Type : MCQ Calculator : Yes Correct Marks : 4

Question Label : Multiple Choice Question

Consider the Java code given below.

```
class Oven {
    public void heatingMethod() {
        System.out.println("Standard heating");
    }

    public void capacity() {
        System.out.println("Standard capacity");
    }
}
class Microwave extends Oven {
    public void heatingMethod() {
        System.out.println("Microwave heating");
    }
}
class ConvectionOven extends Oven {
    public void heatingMethod() {
        System.out.println("Convection heating");
    }
    public void capacity() {
        System.out.println("Large capacity");
    }
}
public class Test {
    static void show(Oven[] ovens) {
        for (int i = 0; i < ovens.length; i++) {
            ovens[i].heatingMethod();
            ovens[i].capacity();
        }
    }
    public static void main(String[] args) {
        Oven[] ovens = {new Microwave(), new ConvectionOven()}; // LINE 1
        show(ovens);
    }
}
```

Choose the correct option.

Options :

6406533043677. ✖

This code generates the output:

```
Convection heating
Large capacity
Convection heating
Large capacity
```

6406533043678. ✖

This code generates the output:

```
Standard heating
Standard capacity
Standard heating
Standard capacity
```

6406533043679. ✔

This code generates the output:

```
Microwave heating
Standard capacity
Convection heating
Large capacity
```

6406533043680. ✖

Compilation error at `LINE 1` because a reference variable of type `Oven` cannot refer to an object of class `ConvectionOven`

Question Number : 304 Question Id : 640653903875 Question Type : MCQ Calculator : Yes Correct Marks : 4

Question Label : Multiple Choice Question

Consider the code given below that checks whether two applicants have same skill. Method `equals` is overridden to compare two `Applicant` objects as follows. If two applicants have the same skill then they are said to be equal. Based on the given information, answer the question that follows.

```
class Applicant {
    private String name;
    private String skill;
    // Constructor to initialize instance variables
    public String toString() {
        return name;
    }
    public boolean equals(Object obj) {
        // CODE BLOCK
    }
}
public class Test {
    public static void main(String[] args) {
        Applicant a1 = new Applicant("Amritha", "Java");
        Applicant a2 = new Applicant("Harshi", "Python");
        Applicant a3 = new Applicant("Kiran", "Java");
        if (a1.equals(a3)) {
            System.out.println(a1 + " and " + a3 + " have the same skill");
        }
        if (a2.equals(a3)) {
            System.out.println(a2 + " and " + a3 + " have the same skill");
        }
    }
}
```

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

```
Amritha and Kiran have the same skill
```

Options :

6406533043681. ✖
```
if(obj instanceof Applicant) {
   if(this.skill.equals(obj.skill))
       return true;
}
return false;
```

6406533043682. ✖
```
if(this.skill.equals(obj.skill))
    return true;
return false;
```

6406533043683. ✔

```
if(obj instanceof Applicant) {
   Applicant a = (Applicant) obj;
   if(this.skill.equals(a.skill))
       return true;
}
return false;
```

```
if(obj instanceof Applicant) {
    Applicant a = obj;
    if(this.skill.equals(a.skill))
        return true;
}
return false;
```

6406533043684.

Question Number : 305 Question Id : 640653903876 Question Type : MCQ Calculator : Yes Correct Marks : 4

Question Label : Multiple Choice Question

Consider the Java code given below.

```
class Secretary {
    private String name;
    public Secretary(String n) {
        name = n;
    }
    public Secretary(Secretary s) {
        this.name = s.name;
    }
    public void setName(String n) {
        name = n;
    }
    public String getName() {
        return name;
    }
}
public class Test {
    public static void main(String[] args) {
        Secretary s1 = new Secretary("Ramesh");
        Secretary s2 = new Secretary(s1);
        Secretary s3 = s1;
        s1.setName("Suresh");
        System.out.println(s1.getName());
        System.out.println(s2.getName());
        System.out.println(s3.getName());
    }
}
```

What will the output be?

Options :

6406533043685. ✖
```
Suresh
Ramesh
Ramesh
```

6406533043686. ✔
```
Suresh
Ramesh
Suresh
```

6406533043687. ✖
```
Suresh
Suresh
Suresh
```

6406533043688. ✖
```
Suresh
Suresh
Ramesh
```

Question Number : 306 Question Id : 640653903877 Question Type : MCQ Calculator : Yes Correct Marks : 4

Question Label : Multiple Choice Question

Consider the code given below.

```
class Book {
    public void open() {
        System.out.println("Book is open");
    }
}
class EBook extends Book {
    public void read() {
        System.out.println("Reading EBook");
    }
}
class AudioBook extends EBook {
    public void listen() {
        System.out.println("Listening to AudioBook");
    }
    public void read() {
        System.out.println("Reading summary");
    }
}

public class TestBook {
    public static void main(String[] args) {
        Book b = new EBook();
        EBook e = new AudioBook(); // LINE 1
        b.open();
        ((EBook)b).read(); // LINE 2
        e.listen();  // LINE 3
    }
}
```

Choose the correct option.

Options :

6406533043689. ✖ LINE 1 generates a compilation error because a variable of type EBook cannot refer to an object of type AudioBook.

6406533043690. ✖ LINE 2 generates a compilation error because a variable of type Book cannot be type cast to an object of type EBook.

6406533043691. ✔

LINE 3 generates a compilation error because the method `listen()` is not defined in class `EBook`.

This code generates the output:

```
Book is open
Reading summary
Listening to AudioBook
```

6406533043692. ✖

Question Number : 307 Question Id : 640653903878 Question Type : MCQ Calculator : Yes Correct Marks : 4

Question Label : Multiple Choice Question

Consider the Java code given below.

```
import java.util.*;
public class Main{
    public static void main(String[] args) {
        ArrayDeque<String> queue1 = new ArrayDeque<String>();
        queue1.add("Orange");
        queue1.addLast("Apple");
        queue1.addFirst("Mango");
        queue1.add("Strawberry");
        queue1.addFirst("Orange");
        System.out.println(queue1);
        TreeSet<String> set = new TreeSet<String>(queue1);
        System.out.println(set);
    }
}
```

What will the output be?

Options :

This program generates the output:

```
[Orange, Mango, Orange, Apple, Strawberry]
[Apple, Mango, Orange, Strawberry]
```

6406533043693. ✔

This program generates the output:

```
[Apple, Mango, Orange, Strawberry]
[Orange, Mango, Orange, Apple, Strawberry]
```

6406533043694. ✖

This program generates the output:

```
[Apple, Mango, Orange, Strawberry]
[Apple, Mango, Orange, Strawberry]
```

6406533043695. ✖

This program generates the output:

```
[Orange, Mango, Orange, Apple, Strawberry]
[Orange, Mango, Orange, Apple, Strawberry]
```

6406533043696.

Question Number : 308 Question Id : 640653903879 Question Type : MCQ Calculator : Yes Correct Marks : 4

Question Label : Multiple Choice Question

Consider the Java code given below.

```
class Actor {
    String name;
    public Actor(String n) {
        this.name = n;
    }
}
class Movie implements Cloneable {
    String title;
    Actor[] actors;
    public Movie(String t, Actor[] a) {
        title = t;
        actors = a;
    }
    public Movie clone() throws CloneNotSupportedException {
        Movie m = (Movie) super.clone();
        m.actors = this.actors.clone();
        return m;
    }
}
public class Test {

     public static void main(String[] args) throws CloneNotSupportedException {
         Actor[] actors1 = { new Actor("Vijay"), new Actor("Nayan") };
         Movie movie1 = new Movie("Action Movie", actors1);
         Movie movie2 = movie1.clone();
         Actor[] actors2 = movie2.actors;
         actors2[0].name = "Surya";
         movie2.title = "Comedy Movie";

         System.out.println(movie1.title + " : " + movie1.actors[0].name);
         System.out.println(movie2.title + " : " + movie2.actors[0].name);
     }
 }
```

What will the output be?

Options :

6406533043697. ✖
```
Comedy Movie : Surya
Comedy Movie : Surya
```

6406533043698. ✖
```
Action Movie : Surya
Action Movie : Surya
```

6406533043699. ✖
```
Action Movie : Vijay
Comedy Movie : Surya
```

6406533043700. ✔
```
Action Movie : Surya
Comedy Movie : Surya
```

Question Number : 309 Question Id : 640653903880 Question Type : MCQ Calculator : Yes Correct Marks : 4

Question Label : Multiple Choice Question

Consider the Java code given below.

```
public class Test {
    public static void main(String[] args) {
        List<String> list = new ArrayList<String>();
        list.add("Arun");
        list.add("Arjun");
        list.add("Prakash");
        list.add("Kavitha");
        list.add("Ameera");

        list.stream().takeWhile(s -> s.startsWith("A"))
            .forEach(s -> System.out.print(s + " "));

        System.out.println();

        list.stream().dropWhile(s -> s.startsWith("A"))
            .forEach(s -> System.out.print(s + " "));

    }
}
```

What will the output be?

Options :

6406533043701. ✖
```
Arun Arjun Ameera
Prakash Kavitha
```

6406533043702. ✖
```
Arun Arjun
Prakash Kavitha
```

6406533043703. ✔
```
Arun Arjun
Prakash Kavitha Ameera
```

6406533043704. ✖
```
Arun Arjun Prakash Kavitha Ameera
```

Question Number : 310 Question Id : 640653903883 Question Type : MCQ Calculator : Yes Correct Marks : 4

Question Label : Multiple Choice Question

Consider the Java code given below.

```
1 class ClassOne{
2      public void methodOne(){
3          // ...
4          methodTwo();
5          // ...
6       }
7       public void methodTwo(){
8          // ...
9       }
10 }
11 class ClassTwo{
12     public static void methodThree(){
13         // ...
14         ClassOne c = new ClassOne();
15         c.methodOne();
16         // ...
17     }
18     public static void methodFour(){
19         // ...
20         methodThree();
21         // ...
22     }
23     public static void main(String[] args) {
24         // ...
25         methodFour();
26     }
27 }
```

During the execution of Line 19 in the above code, the activation record of which method is at the top of the stack of activation records?

Options :

6406533043713. ✖ `main`

6406533043714. ✖ `methodOne`

6406533043715. ✖ `methodTwo`

6406533043716. ✖ `methodThree`

6406533043717. ✔ `methodFour`

Question Number : 311 Question Id : 640653903884 Question Type : MCQ Calculator : Yes Correct Marks : 4

Question Label : Multiple Choice Question

Consider the Java code given below.

```
interface FoodService {
    void orderFood();
}
class FoodDeliveryApp {
    public PizzaService getPizzaService() {
        return new PizzaService();
    }
    public BurgerService getBurgerService() {
        return new BurgerService();
    }

    private class PizzaService implements FoodService {
        public void orderFood() {
            System.out.println("Ordering a pizza");
        }
    }
    private class BurgerService implements FoodService {
        public void orderFood() {
             System.out.println("Ordering a burger");
        }
    }
}
public class Test {
  public static void main(String[] args) {
      FoodDeliveryApp f = new FoodDeliveryApp();
      //CODE BLOCK
      obj1.orderFood();
  }
}
```

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

```
Ordering a pizza
```

Options :

6406533043718. ✖ `PizzaService obj1 = new PizzaService();`

6406533043719. ✔ `FoodService obj1 = f.getPizzaService();`

6406533043720. ✖ `PizzaService obj1 = f.getPizzaService();`

6406533043721. ✖ `FoodService obj1 = new PizzaService();`

Question Number : 312 Question Id : 640653903885 Question Type : MCQ Calculator : Yes Correct Marks : 4

Question Label : Multiple Choice Question

Consider the Java code given below.

```
interface WatchStore {
    default void showDetails() {
        System.out.println("Generic watch");
    }
    default void order() {
        System.out.println("Order placed");
    }
}
class DigitalWatch implements WatchStore { //LINE 1
    public void order() {
        System.out.println("Ordered digital watch");
    }
}
class AnalogWatch implements WatchStore {  //LINE 2
    public void showDetails() {
        System.out.println("Analog watch features");
    }
}

public class Test {
    public static void main(String[] args) {
        WatchStore w[] = new WatchStore[2];
        w[0]= new DigitalWatch();
        w[1] = new AnalogWatch();
       for (WatchStore watch : w) {
            watch.showDetails();
            watch.order();
        }
    }
}
```

Choose the correct option.

Options :

6406533043722. ✖ Compilation error at `LINE 1` because method `showDetails()` is not overridden in class `DigitalWatch`

6406533043723. ✖ Compilation error at `LINE 2` because method `order()` is not overridden in class `AnalogWatch`

6406533043724. ✖

This program generates the output:

```
Ordered digital watch
Analog watch features
```

6406533043725. ✔

This program generates the output:

```
Generic watch
Ordered digital watch
Analog watch features
Order placed
```

Question Number : 313 Question Id : 640653903886 Question Type : MCQ Calculator : Yes Correct Marks : 4

Question Label : Multiple Choice Question

Consider the Java code given below.

```
abstract class AirConditioner {
    abstract void cool();
    void adjustTemperature() {        // LINE 1
        System.out.println("Adjusting temperature");
    }
}
class SplitAC extends AirConditioner {
    void cool() {
        System.out.println("Cooling with Split AC");
    }
    void adjustTemperature() {
        System.out.println("Adjust temp of Split AC");
    }
}
class WindowAC extends AirConditioner {
    void cool() {
        System.out.println("Cooling with Window AC");
    }
    void adjustTemperature() {
        System.out.println("Adjust temp of Window AC");
    }
}
public class Test {
    public static void main(String[] args) {
        AirConditioner ac1 = new SplitAC(); // LINE 2
        AirConditioner ac2 = new WindowAC(); // LINE 3
        ac1.adjustTemperature();
        ac1.cool();
        ac2.adjustTemperature();
        ac2.cool();
    }
}
```

Choose the correct option.

Options :

6406533043726. ✖ `LINE 1` generates compilation error because abstract class must contain only abstract methods.

6406533043727. ✖ `LINE 2` and `LINE 3` generate compilation errors because reference variable of type `AirConditioner` cannot store the objects of type `SplitAC` and `WindowAC`.

6406533043728. ✔

This program generates the output:

```
Adjust temp of Split AC
Cooling with Split AC
Adjust temp of Window AC
Cooling with Window AC
```

6406533043729.

This program generates the output:

```
Adjusting temperature
Cooling with Split AC
Adjusting temperature
Cooling with Window AC
```

Question Number : 314 Question Id : 640653903887 Question Type : MCQ Calculator : Yes Correct Marks : 4

Question Label : Multiple Choice Question

Consider the Java code given below.

```
interface Iterator {
    public boolean has_next();
    public Object get_next();
}
abstract class Printable {
    public abstract void print();
}
class RestaurantMenu {
    private final int limit = 3;
    private Dish[] menu = {
        new Dish("Paneer 65", "D1001"),
        new Dish("Biryani", "D1002"),
        new Dish("MasalaDosa", "D1003"),
    };
    private class Dish extends Printable {
        private String name,code;
        // Constructor to initialize instance variables
        public void print() {
            System.out.println( code + ", " + name);
        }
    }
    private class DishIterator implements Iterator{
        private int index;
        public DishIterator() {
            // Constructor
        }

        public boolean has_next() {
            //if next element is available in list return true;
            //else false
        }
        public Object get_next() {
            //return next element from list
        }
        public Iterator getIterator() {
            return new DishIterator();
        }
    }
}
public class IterTest {
   public static void main(String[] args) {
        RestaurantMenu menu = new RestaurantMenu();
        Iterator iter = menu.getIterator();
        while (iter.has_next()) {
            ---------------- ; //LINE 1
        }
    }
}
```

Identify the appropriate statement to fill in the blank at LINE 1, such that the output is:

```
D1001, Paneer 65
D1002, Biryani
D1003, MasalaDosa
```

Options :

6406533043730. ✔ `((Printable)iter.get_next()).print()`

6406533043731. ✖ `((Dish)iter.get_next()).print()`

6406533043732. ✖ `((RestaurantMenu)iter.get_next()).print()`

6406533043733. ✖
```
iter.get_next().print();
```

Question Number : 315 Question Id : 640653903889 Question Type : MCQ Calculator : Yes Correct Marks : 4

Question Label : Multiple Choice Question

Consider the code given below.

```
import java.util.*;
class OddValueExcepteion extends Exception {
    public String toString() {
        return "Odd number encountered during update";
    }
}
public class Test {
    public static void update(int[] array, int index) throws OddValueExcepteion {
        if (array[index] % 2 != 0) {
            throw new OddValueExcepteion();
        }
        array[index] = array[index] * 5;
    }
    public static void main(String[] args) {
        int[] arr = {4, 6, 3, 2, 1};
        try {
            for (int i = 0; i < arr.length; i++) {
                update(arr, i);
            }
        } catch (OddValueExcepteion e) {
            System.out.println(e);
        }
        for (int n : arr) {
            System.out.print(n + " ");
        }
    }
}
```

What will the output be?

Options :

6406533043738. ✖
```
Odd number encountered during update
```

6406533043739. ✖
```
20 30 3 2 1
```

6406533043740. ✖
```
Odd number encountered during update
20 30
```

6406533043741. ✔

```
Odd number encountered during update
20 30 3 2 1
```

Question Number : 316 Question Id : 640653903890 Question Type : MCQ Calculator : Yes 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 Music {
  HashMap<String, String> artists = new HashMap<>();
    public Music() {
      artists.put("Classical", "Ravi");
      artists.put("Ghazal", "Arijit");
    }
    public String getArtist(String genre) {
       return artists.get(genre);
    }
}
public class Test {
   public static void main(String[] args) {
      Optional<String> a1 = Optional.ofNullable(new Music().getArtist("Classical"))
      Optional<String> a2 = Optional.ofNullable(new Music().getArtist("Pop"));
      a1.ifPresent(n -> System.out.println(n.toUpperCase()));
      a2.ifPresent(n -> System.out.println(n.toUpperCase()));
    }
}
```

Choose the correct option.

Options :

6406533043742. ✔ This program generates the output:
`RAVI`

6406533043743. ✖ This program terminates due to `NullPointerException` after printing the message:
`RAVI`

6406533043744. ✖

This program generates the output:

```
RAVI
null
```

This program generates the output:

```
CLASSICAL
RAVI
```

6406533043745. ✖

Question Number : 317 Question Id : 640653903891 Question Type : MCQ Calculator : Yes Correct Marks : 4

Question Label : Multiple Choice Question

Consider the Java code given below.

```
import java.io.*;
class AccessCard implements Serializable {
    private String cardNumber = "******";
    private transient String employeeName = "ABC";
    private String validityPeriod = "00/00";
    public AccessCard(String c, String n, String p) {
        cardNumber = c;
        employeeName = n;
        validityPeriod = p;
    }
    public String toString() {
        return cardNumber + ", " + employeeName + ", " + validityPeriod;
    }
}
public class Test {
    public static void main(String[] args) throws Exception {
        var fos = new FileOutputStream("AccessCard.txt");
        var os = new ObjectOutputStream(fos);
        os.writeObject(new AccessCard("C987654", "Piyush", "12/25"));
        os.close();
        var fis = new FileInputStream("AccessCard.txt");
        var ois = new ObjectInputStream(fis);
        AccessCard card = (AccessCard) ois.readObject();
        ois.close();
        System.out.println(card);
    }
}
```

What will the output be?

Options :

6406533043746. ✖ `null, null, null`

6406533043747. ✔ C987654, null, 12/25

6406533043748. ✖ C987654, ABC, 12/25

6406533043749. ✖ C987654, Piyush, 03/24

6406533043750. ✖ ********, ABC, 00/00

Question Number : 318 Question Id : 640653903892 Question Type : MCQ Calculator : Yes Correct Marks : 4

Question Label : Multiple Choice Question

Consider the Java code given below.

```
import java.util.*;
import java.util.stream.*;
class Motor {
    private String company;
    private double efficiency;
    // Constructor to initialize instance variables
    public double getEfficiency() {
        return efficiency;
    }
    public String toString() {
        return company;
    }
}
public class Test {
    public static void main(String[] args) {
        var motorList = new ArrayList<Motor>();
        motorList.add(new Motor("Kirloskar", 85.5));
        motorList.add(new Motor("Bharat", 92.5));
        motorList.add(new Motor("Crompton", 75.5));
        motorList.add(new Motor("Siemens", 90.0));

        Map<Boolean, List<Motor>> eMap;
        eMap = motorList.stream()
               .collect(Collectors.partitioningBy(m -> m.getEfficiency() >= 90.0));
        System.out.println(eMap.get(false));
    }
}
```

Choose the correct option.

Options :

6406533043751. ✖ This program generates the output: [Bharat, Siemens]

6406533043752. ✖ This program generates the output: [Bharat]

6406533043753. ✔ This program generates the output: [Kirloskar, Crompton]

6406533043754. ✖ This program generates the output: [Kirloskar, Crompton, Siemens]

Question Number : 319 Question Id : 640653903893 Question Type : MCQ Calculator : Yes Correct Marks : 4

Question Label : Multiple Choice Question

Consider the code given below. Assume that the file tree.txt contains the following lines of text in it.

A tree provides shade and beauty.
Trees are vital for the ecosystem.
Trees offer shelter to wildlife.

```
import java.io.*;
import java.util.Scanner;
public class Example {
    public static void main(String[] args) {
        try {
            var in=new FileInputStream("tree.txt");
            var scanner=new Scanner(in);   //LINE 1
            System.out.println("Data from file:");
            System.out.println(scanner.nextLine());
            System.out.println(scanner.next());
            System.out.println(scanner.nextLine());
        }
        catch (FileNotFoundException e) {
            System.out.println("File does not exist.");
        }
        catch (IOException e) {
            System.out.println("Error in writing a file.");
        }
    }
}
```

Choose the correct option.

Options :

6406533043755. ✖ LINE 1 generates IOException.

6406533043756. ✖

This program generates the output:

```
Data from file:
A tree provides shade and beauty.
Trees are vital for the ecosystem.
Trees offer shelter to wildlife.
```

6406533043757. ✔

This program generates the output:

```
Data from file:
A tree provides shade and beauty.
Trees
  are vital for the ecosystem.
```

6406533043758. ✖

This program generates the output:

```
Data from file:
A tree provides shade and beauty.
A
Trees offer shelter to wildlife.
```

Sub-Section Number : 3

Sub-Section Id : 640653134047

Question Shuffling Allowed : Yes

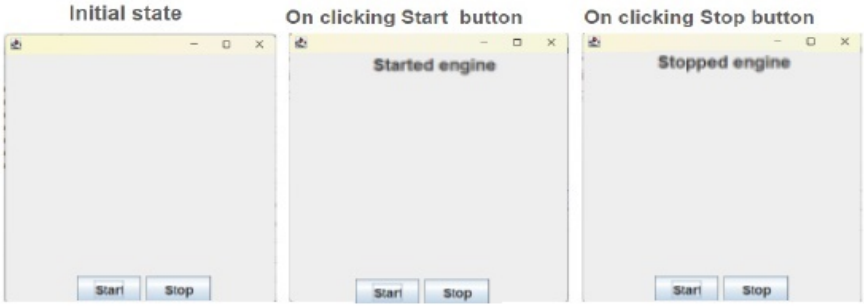
Question Number : 320 Question Id : 640653903881 Question Type : MSQ Calculator : Yes
Correct Marks : 6 Max. Selectable Options : 0

Question Label : Multiple Select Question

Consider the Java code given below.

```
import javax.swing.*;
import java.awt.event.*;
public class ButtonEventTest extends JFrame implements ActionListener{
    private JButton b1, b2;
    private JLabel l1;
    JPanel panel1, panel2;
    public ButtonEventTest() {
        b1 = new JButton("Start");
        b2 = new JButton("Stop");
        panel1 = new JPanel();
        panel1.add(b1);
        panel1.add(b2);
        add(panel1, "South");

        l1 = new JLabel("");
        panel2 = new JPanel();
        panel2.add(l1);
        add(panel2, "North");

        setVisible(true);
        setSize(400, 400);

        b1.setActionCommand("action1");
        b2.setActionCommand("action2");

        b1.addActionListener(this);
        b2.addActionListener(this);
    }
    public void actionPerformed(ActionEvent e) {
        //CODE SEGMENT
    }
    public static void main(String[] args) {
        new ButtonEventTest();
    }
}
```

Choose the correct code segment(s) to be filled inside method `actionPerformed()` such that on clicking the `Start` button, the label text changes to `Started engine` and on clicking the `Stop` button, the label text changes to `Stopped engine`.



Options :

```
if(e.getActionCommand().equals("action1"))
    l1.setText("Started engine");
else if(e.getActionCommand().equals("action2"))
    l1.setText("Stopped engine");
```

6406533043705. ✔

6406533043706. ✖

```
if(e.getActionCommand().equals("b1"))
    l1.setText("Started engine");
else if(e.getActionCommand().equals("b2"))
    l1.setText("Stopped engine");
```

6406533043707. ✔

```
if(e.getSource().equals(b1))
    l1.setText("Started engine");
else if(e.getSource().equals(b2))
    l1.setText("Stopped engine");
```

6406533043708. ✖

```
if(e.getSource().equals("action1"))
    l1.setText("Started engine");
else if(e.getSource().equals("action2"))
    l1.setText("Stopped engine");
```

Question Number : 321 Question Id : 640653903894 Question Type : MSQ Calculator : Yes Correct Marks : 6 Max. Selectable Options : 0

Question Label : Multiple Select Question

Consider the Java code given below.

```
class Pattern implements Runnable {
    boolean stopRequested = false;
    String[] pattern = {"A", "AA", "AAA", "AAAA", "AAAAA"};
    int index = 0;
    public void run() {
        while (!stopRequested) {
            System.out.print(pattern[index] + " ");
            index = (index + 1) % pattern.length;
        }
    }
    public void setStop(boolean stop) {
        stopRequested = stop;
    }
}
public class Test {
    public static void main(String[] args) throws InterruptedException {
        Pattern p = new Pattern();
        Thread t1 = new Thread(p);
        t1.start();
        p.setStop(true);
    }
}
```

Choose the correct option(s).

Options :

6406533043759. ✖ The program will always generate the output: `A AA AAA AAAA AAAAA`

6406533043760. ✖ The program will always generate the output: `A`

6406533043761. ✔ The output can be `A` or `A AA` or `A AA AAA` or `A AA AAA AAAA` or `A AA AAA AAAA AAAAA` and can also cycle back and start again.

6406533043762. ✔ The program may not generate any output.

Sub-Section Number : 4
Sub-Section Id : 640653134048
Question Shuffling Allowed : Yes

Question Number : 322 Question Id : 640653903882 Question Type : MCQ Calculator : Yes Correct Marks : 5
Question Label : Multiple Choice Question

Consider the Java code given below.

```
import javax.swing.*;
import java.awt.*;
public class GUITest extends JFrame {
    JPanel pnlLbl, pnlTxt, pnlBtn;
    JLabel lblId, lblPwd;
    JTextField txtId, txtPwd;
    JButton btn;
    public GUITest() {
        lblId = new JLabel("DiscourseID:");
        lblPwd = new JLabel("password:");
        txtId = new JTextField(10);
        txtPwd = new JTextField(10);
        btn = new JButton("Login");
        pnlLbl = new JPanel();
        //add lblId and txtId to pnlLbl
        pnlTxt = new JPanel();
        //add lblPwd and txtPwd to pnlTxt
        pnlBtn = new JPanel();
        //add btn to pnlBtn

        //CODE BLOCK

        setVisible(true);
        setSize(300,200);
    }
    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.

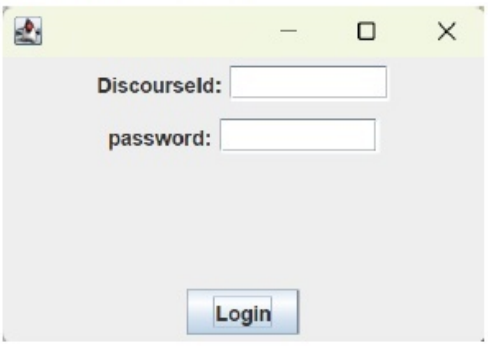


Figure 1

Options :

6406533043709. ✖

```
add(pnlLbl, "Center");
add(pnlTxt, "North");
add(pnlBtn, "South");
```

6406533043710. ✖

```
add(pnlLbl, "North");
add(pnlTxt, "South");
add(pnlBtn, "Center");
```

6406533043711.

```
add(pnlLbl, "North");
add(pnlTxt, "Center");
add(pnlBtn, "South");
```

6406533043712.

```
add(pnlLbl, "South");
add(pnlTxt, "Center");
add(pnlBtn, "North");
```

Sub-Section Number : 5
Sub-Section Id : 640653134049
Question Shuffling Allowed : Yes

Question Number : 323 Question Id : 640653903888 Question Type : MSQ Calculator : Yes Correct Marks : 5 Max. Selectable Options : 0

Question Label : Multiple Select Question

Consider the Java code given below that prints the highest salaried employee type among a set of given EmpType objects. From among the options, identify the appropriate function header for the function printHighestSalariedEmpType that takes as input an array of EmpType objects and prints the highest salaried employee type.

```
import java.util.*;
interface EmpType {
    public abstract double getSalary();
}
class Manager implements EmpType {
    private double salary;
    // Constructor
    // Method to getSalary() that returns salary
}
class Developer implements EmpType {
    private double salary;
    // Constructor
    // Method to getSalary() that returns salary
}

public class Test {
    //LINE 1 : FUNCTION HEADER
    {
      //invokes method getSalary()
      //to print the value of highest priced stock
    }
     public static void main(String[] args) {
         EmpType[] e = {
             new Manager(85000.0),
             new Developer(95000.0),
             new Manager(90000.0)
         };
         printHighestSalariedEmpType(e);
     }
 }
```

Choose the correct option(s).

Options :

6406533043734. ✖ `public static <T extends Manager> void printHighestSalaryEmpType(T[] e)`

6406533043735. ✔ `public static <T extends EmpType> void printHighestSalaryEmpType(T[] e)`

6406533043736. ✖ `public static <T extends Developer> void printHighestSalaryEmpType(T[] e)`

6406533043737. ✔ `public static void printHighestSalaryEmpType(EmpType[] e)`

Question Number : 324 Question Id : 640653903895 Question Type : MSQ Calculator : Yes Correct Marks : 5 Max. Selectable Options : 0

Question Label : Multiple Select Question

Consider the Java code given below.

```
class Hostel {
    int available = 1;
    public synchronized void bookSeat(int n, String name) {
        if (available >= n) {
            available = available - n;
            System.out.println(name + " booked " + n + " seat");
        } else {
            System.out.println(name + " cannot book " + n + " seat");
        }
    }
}
class SeatBooking implements Runnable {
    private Hostel h;
    private String name;
    private int n_seats;
    public SeatBooking(Hostel h, String n, int ns) {
        this.h = h;
        this.name = n;
        this.n_seats = ns;
    }
    public void run() {
        h.bookSeat(n_seats, name);
    }
}
 public class Test {
     public static void main(String[] args) {
         Hostel obj = new Hostel();
         SeatBooking sb1 = new SeatBooking(obj, "Manju", 1);
         SeatBooking sb2 = new SeatBooking(obj, "Parvati", 1);
         Thread t1 = new Thread(sb1);
         Thread t2 = new Thread(sb2);
         t1.start();
         t2.start();
     }
 }
```

Which of the following options is/are possible result/s of the above code?

Options :

6406533043763. ✔

```
Manju booked 1 seat
Parvati cannot book 1 seat
```

6406533043764. ✖

```
Manju booked 1 seat
Parvati booked 1 seat
```

6406533043765. ✔

```
Parvati booked 1 seat
Manju cannot book 1 seat
```

6406533043766. ✖

```
Manju cannot book 1 seat
Parvati cannot book 1 seat
```

Question Number : 325 Question Id : 640653903896 Question Type : MSQ Calculator : Yes Correct Marks : 5 Max. Selectable Options : 0

Question Label : Multiple Select Question

Consider the Java code given below.

```
import java.util.*;
import java.util.concurrent.*;
class Example extends Thread {
    Map cuMap;
    Example(Map m) {
        this.cuMap = m;
    }
    public void run() {
        cuMap.put("4","Hari");
    }
}
public class Test {
    public static void main (String[] args) {
        Map<Integer, String> cuMap = new ConcurrentHashMap();
        Integer[] iarr = {1, 2, 3};
        String[] arr = {"Anjali", "Bharathan", "Dilsha"};
        for(int i = 0; i < iarr.length; i++) {
            cuMap.put(iarr[i],arr[i]);
        }
        Example t = new Example(cuMap);
        t.start();
        Set s = cuMap.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 **true** about the given code.

Options :

6406533043767. ✖ This program may generate `ConcurrentModificationException`.

6406533043768. ✔ The program may generate the output:

```
1 => Anjali
2 => Bharathan
3 => Dilsha
4 => Hari
```

6406533043769. ✔

The program may generate the output:

```
1 => Anjali
2 => Bharathan
3 => Dilsha
```

The program always generate the output:

```
1 => Anjali
2 => Bharathan
3 => Dilsha
4 => Hari
```

6406533043770.

MLT

Section Id :	64065364125
Section Number :	12
Section type :	Online
Mandatory or Optional :	Mandatory
Number of Questions :	18
Number of Questions to be attempted :	18
Section Marks :	50
Display Number Panel :	Yes
Section Negative Marks :	0
Group All Questions :	No
Enable Mark as Answered Mark for Review and Clear Response :	No
Maximum Instruction Time :	0
Sub-Section Number :	1
Sub-Section Id :	640653134050
Question Shuffling Allowed :	No

Question Number : 326 Question Id : 640653903897 Question Type : MCQ Calculator : Yes 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 REGISTERED BY YOU)