

6406532040491. ✖ L = [4, 2, 1, 0, 5, 3]

6406532040492. ✔ L = [4, 2, 0, 1, 3, 5]

6406532040493. ✖ L = [4, 2, 0, 1, 5, 3]

6406532040494. ✖ L = [4, 2, 1, 0, 3, 5]

Question Number : 74 Question Id : 640653610990 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 2

Question Label : Multiple Choice Question

What will be the value of **keys(D)** after executing the given pseudocode?

Options :

6406532040495. ✖ L = [4, 2, 1, 0, 5, 3]

6406532040496. ✖ L = [4, 2, 0, 1, 3, 5]

6406532040497. ✖ L = [4, 2, 0, 1, 5, 3]

6406532040498. ✖ L = [4, 2, 1, 0, 3, 5]

6406532040499. ✔ Can not be determined

Java

Section Id :	64065341307
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
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 :	64065388032
Question Shuffling Allowed :	No
Is Section Default? :	null

Question Number : 75 Question Id : 640653610991 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 :

6406532040500.  YES

6406532040501.  NO

Sub-Section Number :	2
Sub-Section Id :	64065388033
Question Shuffling Allowed :	Yes
Is Section Default? :	null

Question Number : 76 Question Id : 640653610992 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 `Stream.iterate(e, f)` returns an infinite sequential ordered `Stream` produced by iterative application of a function `f` to an initial element `e`, producing a `Stream` consisting of `e`, `f(e)`, `f(f(e))`, etc. Based on the above information, consider the code given below, and answer the question that follows.

```
import java.util.stream.*;
public class Test {
    public static void main(String[] args) {
        Stream.iterate(0, n -> n+3).limit(6).filter(n -> n % 2 == 0)
            .forEach(num -> System.out.print(num+" "));
    }
}
```

What will the output be?

Options :

6406532040502. ✖ 0 2 4 6 8 10

6406532040503. ✔ 0 6 12

6406532040504. ✖ 2 4 6 8 10 12

6406532040505. ✖ 0 2 4 6

Question Number : 77 Question Id : 640653610993 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 following java code

```
class Worker implements Cloneable{
    String name;
    public Worker(String n) {
        name = n;
    }
    public Worker clone() throws CloneNotSupportedException{
        return (Worker)super.clone();
    }
}

class Work implements Cloneable{
    String job;
    Worker[] wk;
    public Work(String j, Worker[] w) {
        job = j;
        wk = w;
    }
    public Work clone() throws CloneNotSupportedException{
        Work t = (Work)super.clone();
        t.wk = t.wk.clone();
        return t;
    }
}

public class Test {
    public static void main(String[] args) throws CloneNotSupportedException{
        Worker[] w1 = {new Worker("ABC"), new Worker("XYZ")};
        Work obj1 = new Work("Painting", w1);
        Work obj2 = obj1.clone();
        Worker[] w2 = obj2.wk;
        w2[0].name = "MNO";
        obj2.job = "Printing";
        System.out.println(obj1.job + " : " + obj1.wk[0].name);
        System.out.println(obj2.job + " : " + obj2.wk[0].name);
    }
}
```

What will the output be?

Options :

Painting : MNO
Painting : MNO

6406532040506. ✖

Printing : MNO
Printing : MNO

6406532040507. ✖

6406532040508. ✖ Painting : ABC
Printing : MNO

6406532040509. ✔ Painting : MNO
Printing : MNO

**Question Number : 78 Question Id : 640653610994 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 Employee{
    private String name;
    private double salary;
    public Employee() {
        this.salary = 40000.0;
    }
    public Employee(Employee e) {
        this.name = e.name;
        this.salary = e.salary;
    }
    public void setName(String name) {
        this.name = name;
    }
    public String getName() {
        return name;
    }
    public void setSalary(double salary) {
        this.salary = salary;
    }
    public double getSalary() {
        return salary;
    }
}

public class Test{
    public static void main(String[] args) {
        Employee obj1 = new Employee();
        Employee obj2 = obj1;
        Employee obj3 = new Employee(obj1);
        obj1.setSalary(50000.0);
        obj1.setName("Anand");
        obj2.setName("Karthik");
        obj3.setName("Shannu");
        System.out.println(obj1.getName() +": " + obj1.getSalary());
        System.out.println(obj2.getName() +": " + obj2.getSalary());
        System.out.println(obj3.getName() +": " + obj3.getSalary());
    }
}
```

What will the output be?

Options :

6406532040510. ✖ Anand: 50000.0

Anand: 50000.0

Shannu: 40000.0

6406532040511. ✔ Karthik: 50000.0

Karthik: 50000.0

Shannu: 40000.0

6406532040512. ✖ Anand: 50000.0

Karthik: 50000.0

Shannu: 40000.0

6406532040513. ✖ Shannu: 40000.0

Shannu: 40000.0

Shannu: 40000.0

Question Number : 79 Question Id : 640653610996 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 method description given below.

`getOrDefault(Object key, V defaultValue)`: Returns the value to which the specified key is mapped, or `defaultValue` if this map contains no mapping for the key.

```
import java.util.*;
class Actor{
    String name;
    String year;
    int no_of_hits;
    public Actor(String name, String year, int no_of_hits) {
        this.name = name;
        this.year = year;
        this.no_of_hits = no_of_hits;
    }
}

public class TestMap{
    public static void printActors(ArrayList<Actor> aL) {
        var map = new TreeMap<String, Integer>();
        for(Actor a:aL) {
            map.put(a.name, map.getOrDefault(a.name, 0)+a.no_of_hits);
        }
        for (Map.Entry<String, Integer> e:map.entrySet()) {
            System.out.println(e.getKey()+" = "+e.getValue());
        }
    }
    public static void main(String[] args) {
        var aList = new ArrayList<Actor>();
        aList.add(new Actor("Sharukh", "2005", 3));
        aList.add(new Actor("Hrithik", "2006", 1));
        aList.add(new Actor("Hrithik", "2005", 2));
        aList.add(new Actor("Sharukh", "2006", 2));
        printActors(aList);
    }
}
```

What will the output be?

Options :

6406532040518. ✔ Hrithik = 3

Sharukh = 5

6406532040519. ✖ Hrithik = 2

Sharukh = 2

6406532040520. ✖ Sharukh = 3

Hrithik = 1

6406532040521. ✖ Sharukh = 5

Hrithik = 3

Question Number : 80 Question Id : 640653610997 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 TestSet {
    //CODE BLOCK
    public boolean property(Integer runs) {
        if(runs >= 40)
            return false;
        return true;
    }
    public void validate(Map<String, Integer> m) {
        for(Map.Entry<String, Integer> entry:m.entrySet()) {
            if(property(entry.getValue()))
                set1.add(entry.getKey());
            else
                set2.add(entry.getKey());
        }
    }
    public void display() {
        System.out.println(set1);
        System.out.println(set2);
    }
    public static void main(String[] args) {
        var batsmen = new LinkedHashMap<String, Integer>();
        batsmen.put("Andrew", 28);
        batsmen.put("Evon", 97);
        batsmen.put("Morkel", 34);
        batsmen.put("Chandrapaul", 41);
        batsmen.put("Bell", 83);
        batsmen.put("Mayank", 39);
        TestSet obj = new TestSet();
        obj.validate(batsmen);
        obj.display();
    }
}
```

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

[Andrew, Mayank, Morkel]
[Bell, Chandrapaul, Evon]

Options :

6406532040522. ✓ `TreeSet<String> set1 = new TreeSet<String>();`
`TreeSet<String> set2 = new TreeSet<String>();`

6406532040523. ✗ `LinkedHashSet<String> set1 = new LinkedHashSet<String>();`
`LinkedHashSet<String> set2 = new LinkedHashSet<String>();`

6406532040524. ✖

```
HashSet<String> set1 = new HashSet<String>();  
HashSet<String> set2 = new HashSet<String>();
```

6406532040525. ✖

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

Question Number : 81 Question Id : 640653610998 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 following java code

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

class CarRental{
    final static int rent_per_hour = 300;
    public double getRent(int no_of_hours) throws HoursException{
        double rent = 0.0;
        if(no_of_hours < 0)
            throw new HoursException("Please enter valid hours");
        rent = no_of_hours * rent_per_hour;
        return rent;
    }
}

public class TestException {
    public static void main(String[] args) {
        CarRental r1 = new CarRental();
        CarRental r2 = new CarRental();
        try {
            System.out.println(r1.getRent(34));
            System.out.println(r2.getRent(-23));
        }
        catch(HoursException e) {
            System.out.println(e.getMessage());
        }
        finally {
            System.out.println("Execution finished");
        }
    }
}
```

What will the output be?

Options :

6406532040526. ✖ 10200.0

Execution finished

6406532040527. ✖ Please enter valid hours

Execution finished

6406532040528. ✔ 10200.0

Please enter valid hours

Execution finished

6406532040529. ✖ Please enter valid hours

Execution finished

10200.0

Question Number : 82 Question Id : 640653610999 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 following java code

```
class Interest{
    //p = Principal Amount, r = Rate per Annum, t = Time (years)
    private double p, t, r;
    public Interest(double p, double t, double r) {
        this.p = p;
        this.t = t;
        this.r = r;
    }
    public double calculate() {
        assert p * t * r > 0; // assert-1
        assert p > 0; // assert-2
        assert t > 0; // assert-3
        assert r > 0; // assert-4
        return (p * t * r / 100);
    }
}

public class AssertTest {
    public static void main(String[] args) {
        Interest obj = new Interest(5000.00, -1.00, -16.56);
        System.out.println(obj.calculate());
    }
}
```

Identify the first assert statement that throws the `AssertionError` when the class is executed as:

```
java -ea AssertTest
```

Options :

6406532040530. ✖ assert-1

6406532040531. ✖ assert-2

6406532040532. ✔ assert-3

6406532040533. ✖ assert-4

Question Number : 83 Question Id : 640653611000 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 Founder{
    Map<String, String> l = new HashMap<String, String>();
    public Founder() {
        l.put("Lenovo", "Liu Chuanzhi");
        l.put("Samsung", "Lee Byung-chul");
    }
    public String getFounder(String t){
        return l.get(t);
    }
}
public class TestOptional {
    public static void main(String[] args){
        Optional<String> op1 = Optional.ofNullable(new Founder()
                                            .getFounder("Lenovo"));
        Optional<String> op2 = Optional.ofNullable(new Founder()
                                            .getFounder("samsung"));
        op1.ifPresent(n->System.out.println(n.toUpperCase()));
        op2.ifPresent(n->System.out.println(n.toUpperCase()));
    }
}
```

Choose the correct option.

Options :

This program terminates due to NullPointerException after printing the message:

6406532040534. ✖ LIU CHUANZHI

6406532040535. ✓ This program generates the output:
LIU CHUANZHI

6406532040536. ✖ This program generates the output:
LIU CHUANZHI
NULL

6406532040537. ✖ This program generates the output.
LIU CHUANZHI
LEE BYUNG-CHUL

Question Number : 84 Question Id : 640653611001 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 CollectRes {
    public static void main(String[] args) {
        var list=new ArrayList<String>();
        list.add(null);
        list.add("English Willow");
        list.add(null);
        list.add("Kashmiri Willow");
        list.add("Poular willow");
        list.add("Poular willow");
        list.add("Kashmiri Willow");
        list.add("English Willow");

        // CODE BLOCK

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

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

[English Willow, Kashmiri Willow, Poular willow]

Options :

```
var obj= list.stream()
                .map(str -> str != null)
                .collect(Collectors.toCollection(LinkedHashSet<String>::new));
```

6406532040538. ✖

```
var obj= list.stream()
                .limit(3)
                .filter(str -> str != null)
                .collect(Collectors.toCollection(LinkedHashSet<String>::new));
```

6406532040539. ✖

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

6406532040540. ✔

6406532040541. ✖

```
var obj= list.stream()  
    .limit(3)  
    .map(str -> str != null)  
    .collect(Collectors.toCollection(LinkedHashSet<String>::new));
```

Question Number : 85 Question Id : 640653611002 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, before execution of the given code, the files "f1.txt" and "f2.txt" have the following text in them.

JavaScript is a scripting language

```
import java.io.*;  
public class TestFile {  
    public static void main(String[] args) {  
        try {  
            var out = new FileOutputStream("f1.txt", true);  
            var dout = new DataOutputStream(out);  
            dout.writeBytes(", enables you to create dynamically updating content");  
            dout.close();  
  
            var out2 = new FileOutputStream("f2.txt", false);  
            var dout2 = new DataOutputStream(out2);  
            dout2.writeBytes(", enables you to create dynamically updating content")  
  
            dout2.close();  
        }  
        catch(IOException e) {  
            System.out.println(e);  
        }  
    }  
}
```

Choose the correct option regarding the contents of f1.txt and f2.txt after the program finishes execution.

Options :

6406532040542. ✖

f1.txt:
, enables you to create dynamically updating content
f2.txt:
JavaScript is a scripting language, enables you to create dynamically updating content

f1.txt and f2.txt
JavaScript is a scripting language, enables you to create dynamically updating content
6406532040543. ✖

f1.txt:
JavaScript is a scripting language, enables you to create dynamically updating content
f2.txt:
, enables you to create dynamically updating content
6406532040544. ✔

f1.txt and f2.txt
, enables you to create dynamically updating content
6406532040545. ✖

Question Number : 86 Question Id : 640653611003 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 Coupon implements Serializable{
    private String c_type="****";
    private transient String code="****";
    private String exp="****";
    public Coupon(String type, String c, String e) {
        c_type = type;
        code = c;
        exp = e;
    }
    public String toString() {
        return c_type + ", " + code + ", " + exp;
    }
}
public class STest{
    public static void main(String[] args) throws Exception{
        var fos = new FileOutputStream("coupon.txt");
        var os = new ObjectOutputStream(fos);
        os.writeObject(new Coupon("Accessories", "ACCNEW", "12/08/23"));
        var fis = new FileInputStream("coupon.txt");
        var ois = new ObjectInputStream(fis);
        Coupon r = (Coupon)ois.readObject();
        System.out.println(r);
    }
}
```

What will the output be?

Options :

6406532040546. ✖ ****, ****, ****

6406532040547. ✖ Accessories, ACCNEW, 12/08/23

6406532040548. ✔ Accessories, null, 12/08/23

6406532040549. ✖ null, null, null

6406532040550. ✖ Accessories, ****, 12/08/23

Question Number : 87 Question Id : 640653611006 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.ArrayList;
class Director{
    String name;
    ArrayList<String> awards;
    public Director(String name, ArrayList<String> awards) {
        this.name = name;
        this.awards = awards;
    }
    public Director(Director d){
        name = d.name;
        awards = new ArrayList<String>();
        for(String s : d.awards)
            awards.add(s);
    }
    public String toString() {
        return name+", "+awards;
    }
}

public class TestCon {
    public static void main(String[] args) {
        ArrayList<String> awd = new ArrayList<String>();
        awd.add("Nandi");
        awd.add("Filmfare");
        Director d1 = new Director("Rajamouli", awd);
        Director d2 = new Director(d1);
        d2.name = "Trivikram";
        d1.awards.add("IIFA");
        System.out.println(d1);
        System.out.println(d2);
    }
}
```

What will the output be?

Options :

6406532040559. ✖ Rajamouli, [Nandi, Filmfare]
Trivikram, [Nandi, Filmfare]

6406532040560. ✖ Rajamouli, [Nandi, Filmfare, IIFA]
Trivikram, [Nandi, Filmfare, IIFA]

6406532040561. ✔ Rajamouli, [Nandi, Filmfare, IIFA]
Trivikram, [Nandi, Filmfare]

6406532040562. ✖ Rajamouli, [Nandi, Filmfare]
Trivikram, [Nandi, Filmfare, IIFA]

Question Number : 88 Question Id : 640653611007 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

```
interface Mobile{
    abstract void features();
}
class Mobile4G implements Mobile{
    public void features() {
        System.out.println("Latency ranges from 60 ms to 98 ms");
    }
}
class Mobile5G extends Mobile4G{
    public void features() {
        System.out.println("Latency under 5 milliseconds");
    }
}
public class TestGeneric {
    // LINE 1 : FUNCTION HEADER {
        obj.features();
    }
    public static void main(String[] args) {
        getDetails(new Mobile4G());
        getDetails(new Mobile5G());
    }
}
```

Choose the correct option(s) to be filled in place of LINE 1 so that the program generates the output:

Latency ranges from 60 ms to 98 ms
Latency under 5 milliseconds

Options :

6406532040563. ✖ public static void getDetails(T obj)

6406532040564. ✖ public static <T> void getDetails(T obj)

6406532040565. ✔ public static <T extends Mobile> void getDetails(T obj)

6406532040566. ✖ public static <T implements Mobile> void getDetails(T obj)

Question Number : 89 Question Id : 640653611008 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 three Java files given below.

TestPack1.java:

```
package com.pack1;
public class TestPack1 {
    protected void go() {
        System.out.println("go from TestPack1");
    }
}
```

TestPack2.java:

```
package com.pack2;
public class TestPack2 extends com.pack1.TestPack1{ // LINE-1
    public void move() {
        super.go(); // LINE-2
        System.out.println("move from TestPack2");
    }
    protected void present() {
        System.out.println("present from TestPack2");
    }
}
```

Test.java:

```
package com.pack2;
public class Test {
    public static void main(String[] args) {
        var obj = new com.pack2.TestPack2();
        obj.move(); // LINE-3
        obj.present(); // LINE-4
    }
}
```

Choose the correct option.

Options :

LINE-1 generates a compilation error because of incorrect way of importing the package.

6406532040567. ✖

6406532040568. ✖ LINE-2 generates a compilation error because of illegal access to `go()`.

6406532040569. ✖ LINE-3 generates a compilation error because of illegal access to `move()`.

6406532040570. ✖ LINE-4 generates a compilation error because of illegal access to `present()`.

This program generates the output:

```
go from TestPack1  
move from TestPack2  
present from TestPack2
```

6406532040571. ✔

Question Number : 90 Question Id : 640653611010 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.concurrent.*;
class Example extends Thread{
    Map siMap;
    Example(Map m){
        this.siMap = m;
    }
    public void run(){
        siMap.put("D",4);
    }
}

public class Test{
    public static void main (String[] args) throws InterruptedException{
        Map<String, Integer> siMap = new LinkedHashMap<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();
        t.join();
        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 true about the given code?

Options :

6406532040576. ✖ This program may generate ConcurrentModificationException.

This program generates the output:

A => 1

B => 2

C => 3

D => 4

6406532040577. ✔

6406532040578. ✖ This program results in a deadlock.

This program generates the output:

A => 1
B => 2
C => 3

6406532040579. ✖

Question Number : 91 Question Id : 640653611013 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 and select the correct option from among the given choices.

```
interface Flyable{
    public void fly();
}
interface Moveable extends Flyable{
    public void move();
    public void run();
}
class Flight implements Moveable{
    public void fly(){
        System.out.println("Flight is flying");
    }
    public void move(){
        System.out.println("Flight is moving");
    }
}
public class Test {
    public static void main(String[] args) {
        Flight obj= new Flight();
        obj.fly();
        obj.move();
    }
}
```

Options :

6406532040592. ✖ This code generates the output: Flight is flying

6406532040593. ✔ Compilation fails because run() method has not been overridden in class Flight.

6406532040594. ✖ This code generates the output: Flight is moving

6406532040595. ✖ This code generates the output:
Flight is flying
Flight is moving

Sub-Section Number :	3
Sub-Section Id :	64065388034
Question Shuffling Allowed :	Yes
Is Section Default? :	null

Question Number : 92 Question Id : 640653611004 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

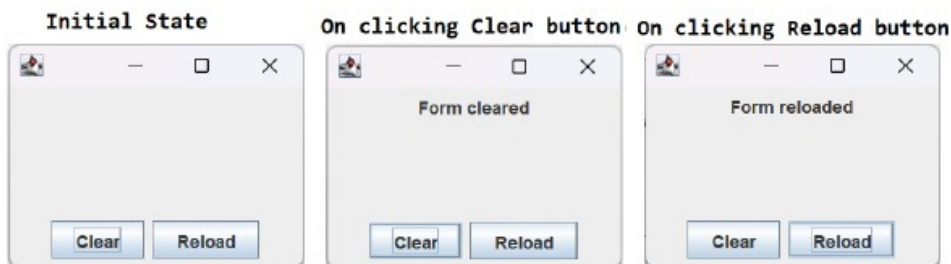
Question Label : Multiple Select Question

Consider the Java program 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("Clear");
        b2=new JButton("Reload");

        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 **Clear** button, the label text changes to **Form cleared** and on clicking the **Reload** button, the label text changes to **Form reloaded**.

Options :

```
if(e.getActionCommand().equals("action1"))
    l1.setText("Form cleared");
else if(e.getActionCommand().equals("action2"))
    l1.setText("Form reloaded");
```

6406532040551. ✓

6406532040552. ✗


```
if(e.getActionCommand().equals("b1"))  
    l1.setText("Form cleared");  
else if(e.getActionCommand().equals("b2"))  
    l1.setText("Form reloaded");
```

```
        if(e.getSource().equals(b1))  
            l1.setText("Form cleared");  
        else if(e.getSource().equals(b2))  
            l1.setText("Form reloaded");
```

6406532040553. ✓

```
        if(e.getSource().equals("action1"))  
            l1.setText("Form cleared");  
        else if(e.getSource().equals("action2"))  
            l1.setText("Form reloaded");
```

6406532040554. ✖

Question Number : 93 Question Id : 640653611011 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 following code

```
class BookingEnquiry{
    int available = 1;
    public synchronized void request(int n, String name){
        if(available >= n){
            available = available - n;
            System.out.println(name + " booked " + n + " ticket");
        }
        else{
            System.out.println(name + " cannot book " + n + " ticket");
        }
    }
}

class TicketBooking implements Runnable{
    BookingEnquiry e;
    String name;
    int n_tickets;
    public TicketBooking(BookingEnquiry e1, String n, int t){
        e = e1;
        name = n;
        n_tickets = t;
    }
    public void run(){
        e.request(n_tickets, name);
    }
}

public class ThreadTest {
    public static void main(String[] args) {
        BookingEnquiry obj = new BookingEnquiry();
        TicketBooking tb1 = new TicketBooking(obj, "Shannu", 1);
        TicketBooking tb2 = new TicketBooking(obj, "Karthik", 1);
        Thread t1 = new Thread(tb1);
        Thread t2 = new Thread(tb2);
        t1.start();
        t2.start();
    }
}
```

Which of the following is/are possible output(s)?

Options :

6406532040580. ✖ Shannu booked 1 ticket
Karthik booked 1 ticket

6406532040581. ✖ Karthik booked 1 ticket
Shannu booked 1 ticket

6406532040582. ✔ Shannu booked 1 ticket

Karthik cannot book 1 ticket

6406532040583. ✔ Karthik booked 1 ticket

Shannu cannot book 1 ticket

6406532040584. ✖ Karthik cannot book 1 ticket

Shannu cannot book 1 ticket

6406532040585. ✖ Shannu cannot book 1 ticket

Karthik cannot book 1 ticket

Question Number : 94 Question Id : 640653611014 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.

```
interface UpGradable{
    void upgrade();
}
class PC {
    public UpGradable getHardware() {
        return new Hardware();
    }
    public UpGradable getOS() {
        return new OS();
    }
    private class Hardware implements UpGradable{
        public void upgrade() {
            System.out.println("It is upgradable");
        }
    }
    private class OS implements UpGradable{
        public void upgrade() {
            System.out.println("It is upgradable");
        }
    }
}

public class PrivateTest {
    public static void main(String[] args) {
        // CODE BLOCK
    }
}
```

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

It is upgradable

It is upgradable

Options :

```
PC obj1 = new PC();
obj1.getHardware().upgrade();
obj1.getOS().upgrade();
```

6406532040596. ✓

```
PC obj1 = new PC();
PC.HardWare = obj1.getHardware();
PC.OS priobj2 = obj1.getOS();
priobj1.upgrade();
priobj2.upgrade();
```

6406532040597. ✗

6406532040598. ✖

```
PC obj1 = new PC();
Hardware priobj1 = obj1.getHardware();
OS priobj2 = obj1.getOS();
priobj1.upgrade();
priobj2.upgrade();
```

6406532040599. ✔

```
PC obj1 = new PC();
UpGradable priobj1 = obj1.getHardware();
UpGradable priobj2 = obj1.getOS();
priobj1.upgrade();
priobj2.upgrade();
```

Sub-Section Number :	4
Sub-Section Id :	64065388035
Question Shuffling Allowed :	Yes
Is Section Default? :	null

Question Number : 95 Question Id : 640653611005 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 Java code given below.

```
import java.util.*;
public class TestQueue{
    public static void main(String[] args) {
        PriorityQueue<String> queue1 = new PriorityQueue<String>();
        queue1.add("Gurgaon");
        queue1.add("Calicut");
        queue1.add("Udaipur");
        queue1.add("Ooty");
        queue1.add("Hyderabad");

        ArrayDeque<String> queue2 = new ArrayDeque<String>();
        while(queue1.size()>0) {
            queue2.addFirst(queue1.poll()); //LINE-1
        }
        System.out.print(queue2);
    }
}
```

Choose the correct option.

Options :

6406532040555. ✖ This program generates the output:
[Hyderabad, Ooty, Udaipur, Calicut, Gurgaon]

6406532040556. ✖ This program generates the output:
[Calicut, Gurgaon, Hyderabad, Ooty, Udaipur]

6406532040557. ✖ Compilation error at LINE-1

6406532040558. ✔ This program generates the output:
[Udaipur, Ooty, Hyderabad, Gurgaon, Calicut]

Question Number : 96 Question Id : 640653611009 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.

```
import java.util.*;
public class TestHigh {
    public static void main(String[] args) {
        List<String> list = new ArrayList<String>();
        list.add("RAT");
        list.add("SKY");
        list.add("FLY");
        list.add("EAGLE");
        list.add("LEMON");
        list.add("VK");
        list.add("MSD");

        list.stream().takeWhile(s->s.length()==3)
            .forEach(s -> System.out.print(s+" "));

        System.out.println();

        list.stream().dropWhile(s->s.length()==3)
            .forEach(s -> System.out.print(s+" "));
    }
}
```

What will the output be?

Options :

RAT SKY FLY MSD

6406532040572. ✖ EAGLE LEMON

RAT SKY FLY

6406532040573. ✖ EAGLE LEMON

RAT SKY FLY

6406532040574. ✔ EAGLE LEMON VK MSD

RAT SKY FLY

6406532040575. ✖ VK

Sub-Section Number : 5
Sub-Section Id : 64065388036
Question Shuffling Allowed : Yes
Is Section Default? : null

Question Number : 97 Question Id : 640653610995 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 statements given below.
Statement 1: Player is an object that has methods doBowling(), doBatting() and doFiledng().
Statement 2: Captain is an object that has methods doBowling(), doBatting(), doFiledng() and doCaptaincy()
Identify the correct option regarding subtyping with respect to Player objects and Captain objects.

Options :

- 6406532040514. ✓ Captain can be a subtype of Player.
- 6406532040515. ✗ Player can be a subtype of Captain.
- 6406532040516. ✗ Captain cannot be a subtype of Player.
- 6406532040517. ✓ Player cannot be a subtype of Captain.

Question Number : 98 Question Id : 640653611012 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 following java code:

```
1.class Actor{
2.    public void show(){
3.        System.out.println("show Actor name");
4.    }
5.    public void display(){
6.        System.out.println("display Actor details");
7.    }
8.}
9.class Hero extends Actor{
10.    public void show() {
11.        System.out.println("show Hero name");
12.    }
13.}
14.class Director extends Actor, Hero{
15.    public void show() {
16.        System.out.println("show Director name");
17.    }
18.}
19.public class Test {
20.    public static void main(String[] args){
21.        Actor obj= new Actor();
22.        obj.show();
23.        Actor obj1 = new Hero();
24.        Hero obj2 = new Actor();
25.    }
26.}
```

Identify the line/s which has/have error.

Options :

6406532040586. ✖ Line 6

6406532040587. ✔ Line 14

6406532040588. ✖ Line 15

6406532040589. ✖ Line 20

6406532040590. ✖ Line 23

6406532040591. ✔ Line 24

MLT

Section Id :	64065341308
Section Number :	5
Section type :	Online
Mandatory or Optional :	Mandatory
Number of Questions :	17
Number of Questions to be attempted :	17
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 :	64065388037
Question Shuffling Allowed :	No
Is Section Default? :	null

Question Number : 99 Question Id : 640653611015 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