Response Type : Numeric

Evaluation Required For SA : Yes

Show Word Count : Yes

Answers Type : Equal

Text Areas : PlainText

Possible Answers :

2

Java

Section Id :	64065348508
Section Number :	10
Section type :	Online
Mandatory or Optional :	Mandatory
Number of Questions :	16
Number of Questions to be attempted :	16
Section Marks :	50
Display Number Panel :	Yes
Group All Questions :	No
Enable Mark as Answered Mark for Review and Clear Response :	Yes
Maximum Instruction Time :	0
Sub-Section Number :	1
Sub-Section Id :	640653100847
Question Shuffling Allowed :	No
Is Section Default? :	null

Question Number : 145 Question Id : 640653689562 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 :

6406532306774. ✓ YES

6406532306775. ✗ NO

Sub-Section Number :	2
Sub-Section Id :	640653100848
Question Shuffling Allowed :	Yes
Is Section Default? :	null

Question Number : 146 Question Id : 640653689564 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 3

Question Label : Multiple Choice Question

Consider the Java code given below.

```
import java.util.*;

public class SetIteratorTest {
    public static void main(String[] args) {
        var set1 = new HashSet<String>();
        set1.add("Cherry");
        set1.add("Banana");
        set1.add("Apple");
        set1.add("Date");

        var set2 = new TreeSet<String>(set1);
        Iterator<String> it1 = set1.iterator();
        Iterator<String> it2 = set2.iterator();

        while (it1.hasNext()) {
            System.out.println(it1.next());
        }

        while (it2.hasNext()) {
            System.out.println(it2.next());
        }
    }
}
```

Choose the correct option.

Options :

6406532306780. ✖ `it1` will visit elements of `set1` in sorted order.
`it2` will visit elements of `set2` in sorted order.

6406532306781. ✖ `it1` will visit elements of `set1` in the order in which they were inserted.
`it2` will visit elements of `set2` in sorted order.

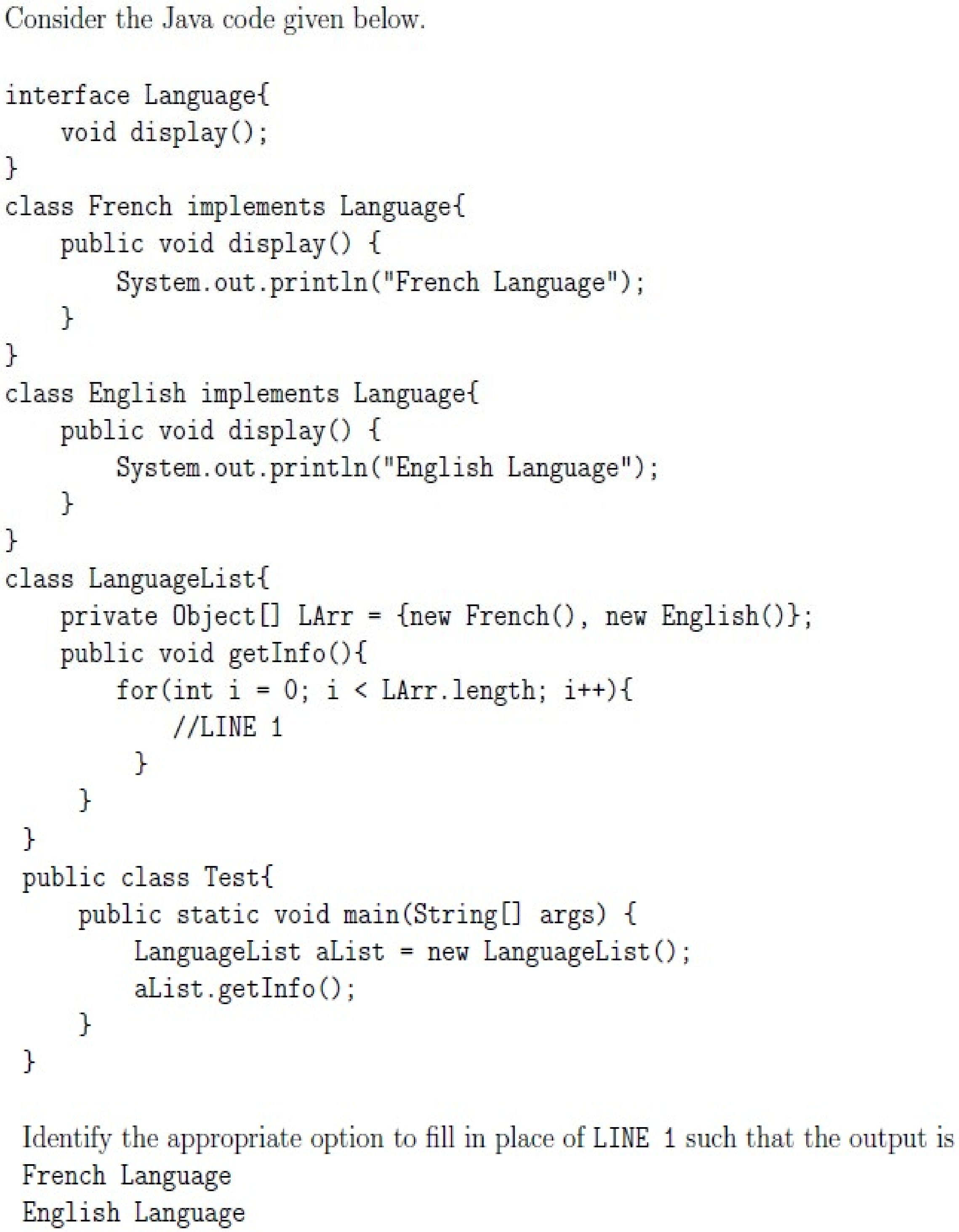
6406532306782. ✓ `it1` will visit elements of `set1` in unspecified order.
`it2` will visit elements of `set2` in sorted order.

6406532306783. ✖ `it1` will visit elements of `set1` in the order in which they were inserted.
`it2` will visit elements of `set2` in unspecified order.

Question Number : 147 Question Id : 640653689567 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 3

Question Label : Multiple Choice Question



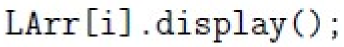
Consider the Java code given below.

```
interface Language{
    void display();
}
class French implements Language{
    public void display() {
        System.out.println("French Language");
    }
}
class English implements Language{
    public void display() {
        System.out.println("English Language");
    }
}
class LanguageList{
    private Object[] LArr = {new French(), new English()};
    public void getInfo(){
        for(int i = 0; i < LArr.length; i++){
            //LINE 1
         }
     }
 }
 public class Test{
     public static void main(String[] args) {
         LanguageList aList = new LanguageList();
         aList.getInfo();
     }
 }
```

Identify the appropriate option to fill in place of LINE 1 such that the output is

```
French Language
English Language
```

Options :

6406532306792. ✖ 

```
LArr[i].display();
```

6406532306793. ✖ `((English)LArr[i]).display();`

6406532306794. ✖ `((French)LArr[i]).display();`

6406532306795. ✔ `((Language)LArr[i]).display();`

Question Number : 148 Question Id : 640653689568 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 3

Question Label : Multiple Choice Question

Consider the following Java code.

```
import java.util.stream.*;

public class Test {
    public static void main(String[] args) {
        String[] colors = {"red", "green", "blue", "yellow"};
        Stream.of(colors).map((color) -> color.length())
            .filter((length) -> length % 2 == 0)
            .forEach((x) -> System.out.println(x));
    }
}
```

What will the output be?

Options :

6406532306796. ✖

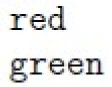
```
3
5
```

6406532306797. ✖

```
blue
yellow
```

6406532306798. ✔

```
4
6
```

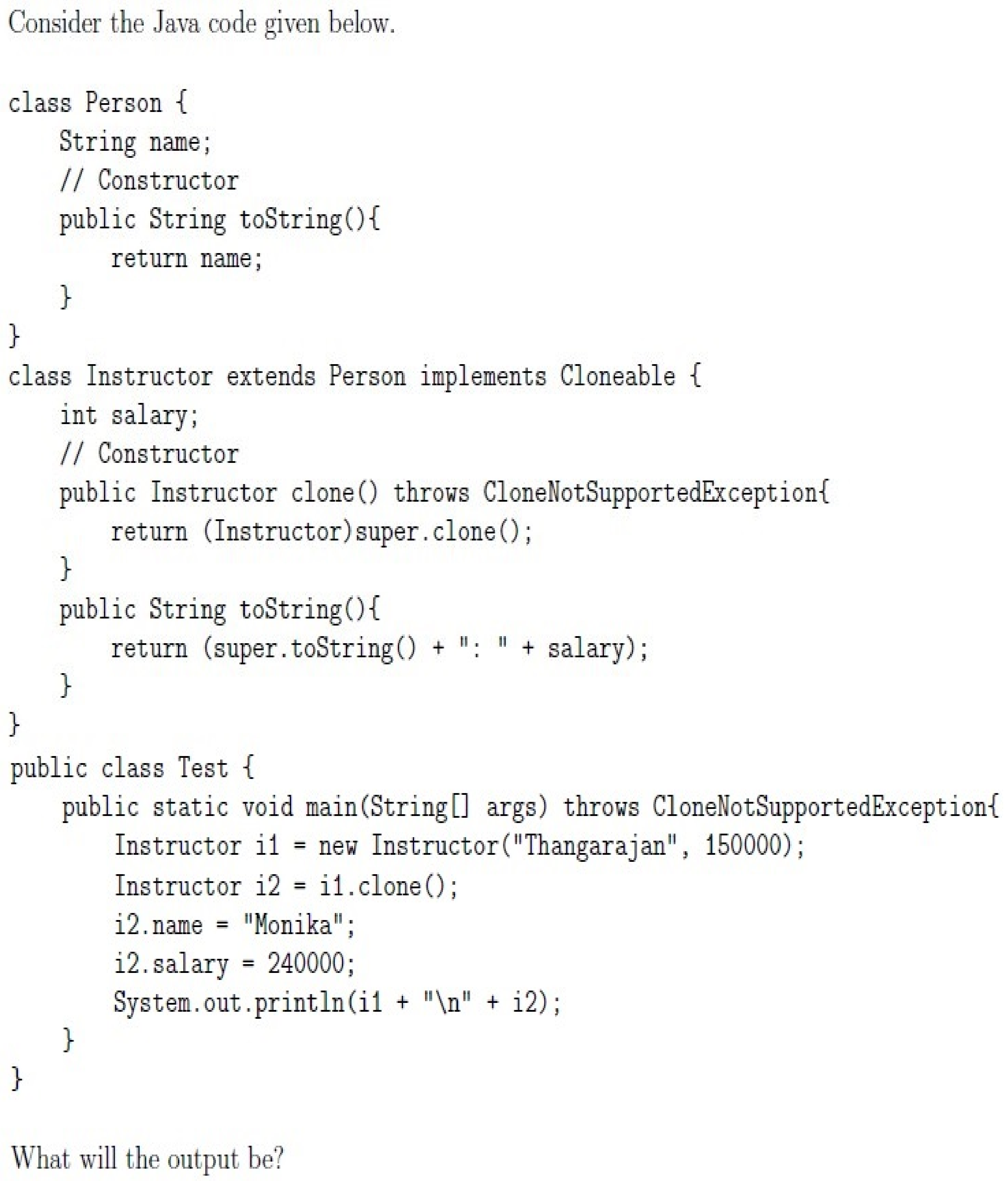
6406532306799. ✖ 

```
red
green
```

Question Number : 149 Question Id : 640653689569 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 3

Question Label : Multiple Choice Question

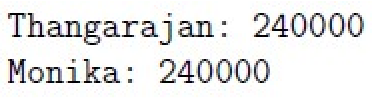


Consider the Java code given below.

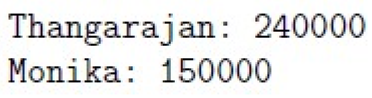
```
class Person {
    String name;
    // Constructor
    public String toString(){
        return name;
    }
}
class Instructor extends Person implements Cloneable {
    int salary;
    // Constructor
    public Instructor clone() throws CloneNotSupportedException{
        return (Instructor)super.clone();
    }
    public String toString(){
        return (super.toString() + ": " + salary);
    }
}
public class Test {
    public static void main(String[] args) throws CloneNotSupportedException{
        Instructor i1 = new Instructor("Thangarajan", 150000);
        Instructor i2 = i1.clone();
        i2.name = "Monika";
        i2.salary = 240000;
        System.out.println(i1 + "\n" + i2);
    }
}
```

What will the output be?

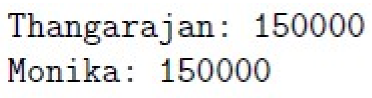
Options :

6406532306800. ✗ 

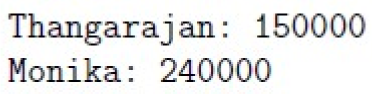
```
Thangarajan: 240000
Monika: 240000
```

6406532306801. ✗ 

```
Thangarajan: 240000
Monika: 150000
```

6406532306802. ✗ 

```
Thangarajan: 150000
Monika: 150000
```

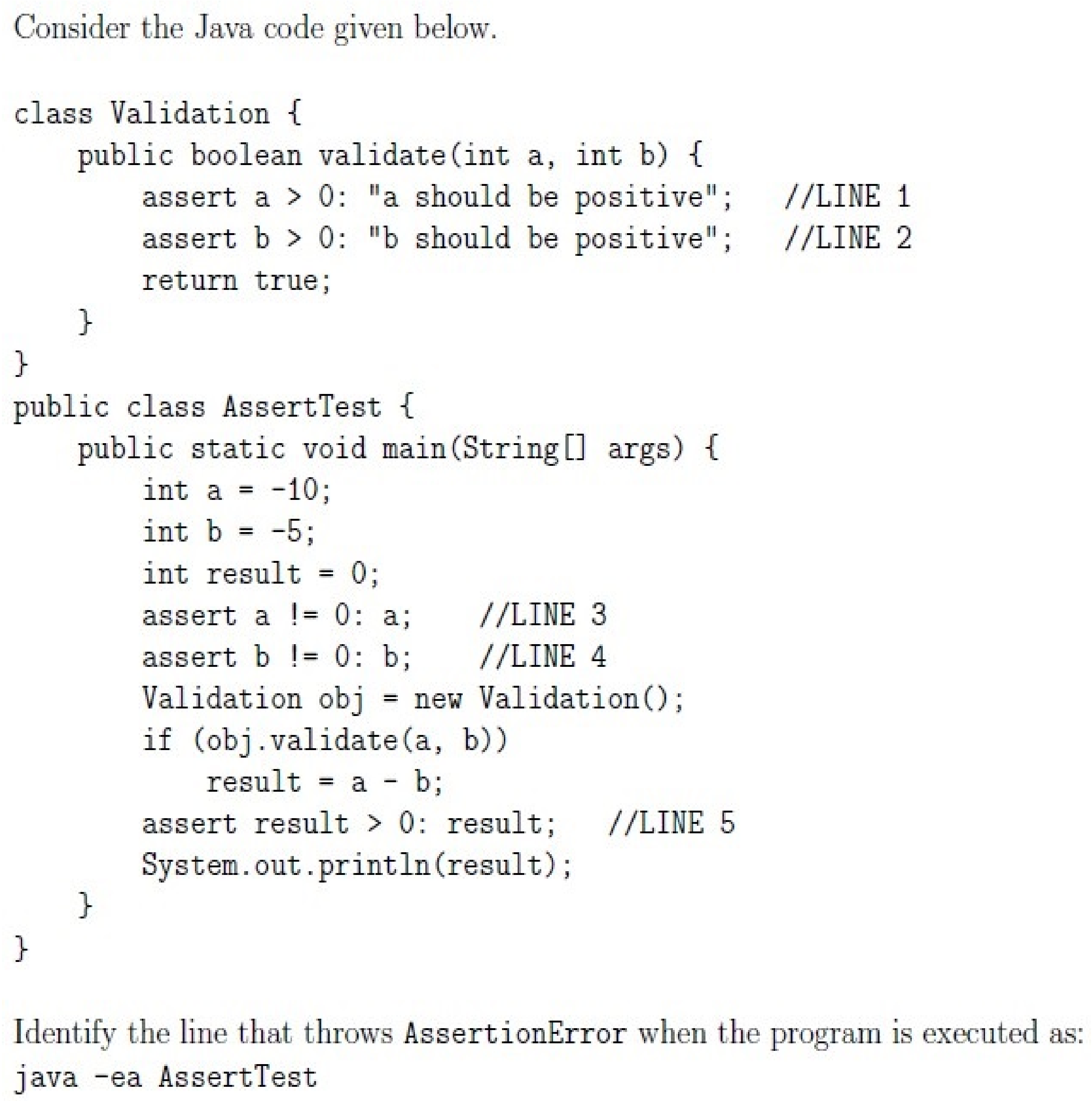
6406532306803. ✓ 

```
Thangarajan: 150000
Monika: 240000
```

Question Number : 150 Question Id : 640653689570 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 3

Question Label : Multiple Choice Question



Consider the Java code given below.

```
class Validation {
    public boolean validate(int a, int b) {
        assert a > 0: "a should be positive";   //LINE 1
        assert b > 0: "b should be positive";   //LINE 2
        return true;
    }
}
public class AssertTest {
    public static void main(String[] args) {
        int a = -10;
        int b = -5;
        int result = 0;
        assert a != 0: a;     //LINE 3
        assert b != 0: b;     //LINE 4
        Validation obj = new Validation();
        if (obj.validate(a, b))
            result = a - b;
        assert result > 0: result;    //LINE 5
        System.out.println(result);
    }
}
```

Identify the line that throws AssertionError when the program is executed as:
`java -ea AssertTest`

Options :

6406532306804. ✓ LINE 1

6406532306805. ✖ LINE 2

6406532306806. ✖ LINE 3

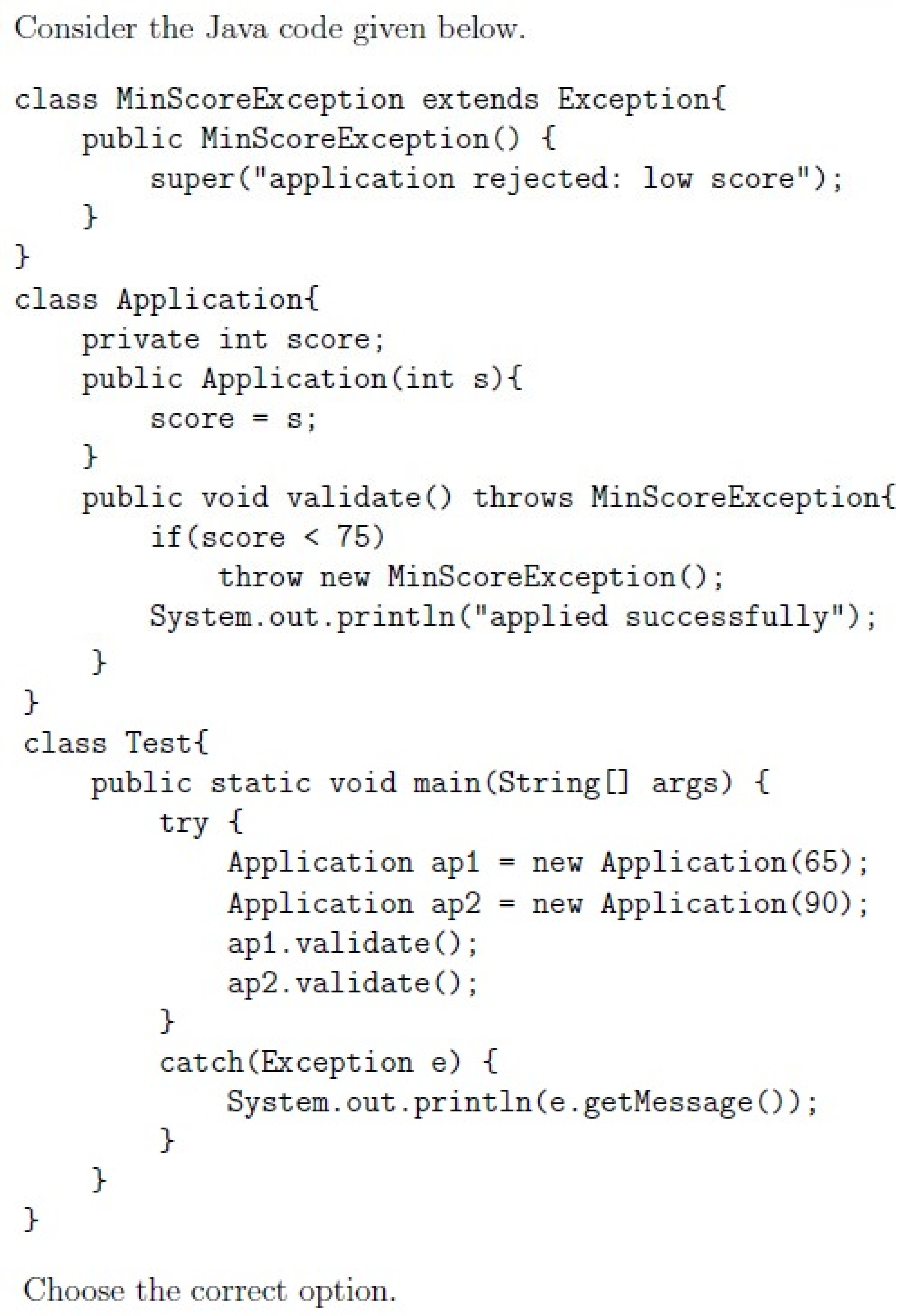
6406532306807. ✖ LINE 4

6406532306808. ✖ LINE 5

Question Number : 151 Question Id : 640653689573 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 3

Question Label : Multiple Choice Question

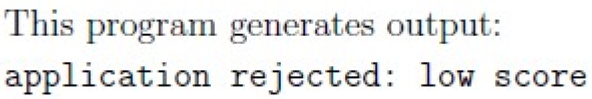


Consider the Java code given below.

```
class MinScoreException extends Exception{
    public MinScoreException() {
        super("application rejected: low score");
    }
}
class Application{
    private int score;
    public Application(int s){
        score = s;
    }
    public void validate() throws MinScoreException{
        if(score < 75)
            throw new MinScoreException();
        System.out.println("applied successfully");
     }
 }
 class Test{
     public static void main(String[] args) {
         try {
             Application ap1 = new Application(65);
             Application ap2 = new Application(90);
             ap1.validate();
             ap2.validate();
         }
         catch(Exception e) {
             System.out.println(e.getMessage());
         }
     }
 }
```

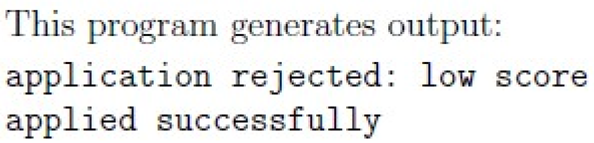
Choose the correct option.

Options :

6406532306817. ✓ 

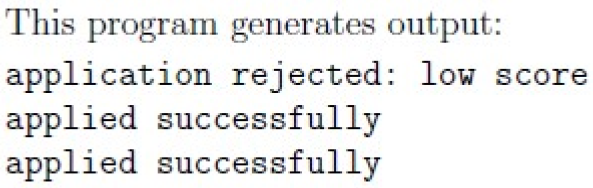
This program generates output:

```
application rejected: low score
```

6406532306818. ✖ 

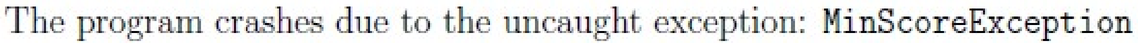
This program generates output:

```
application rejected: low score
applied successfully
```

6406532306819. ✖ 

This program generates output:

```
application rejected: low score
applied successfully
applied successfully
```

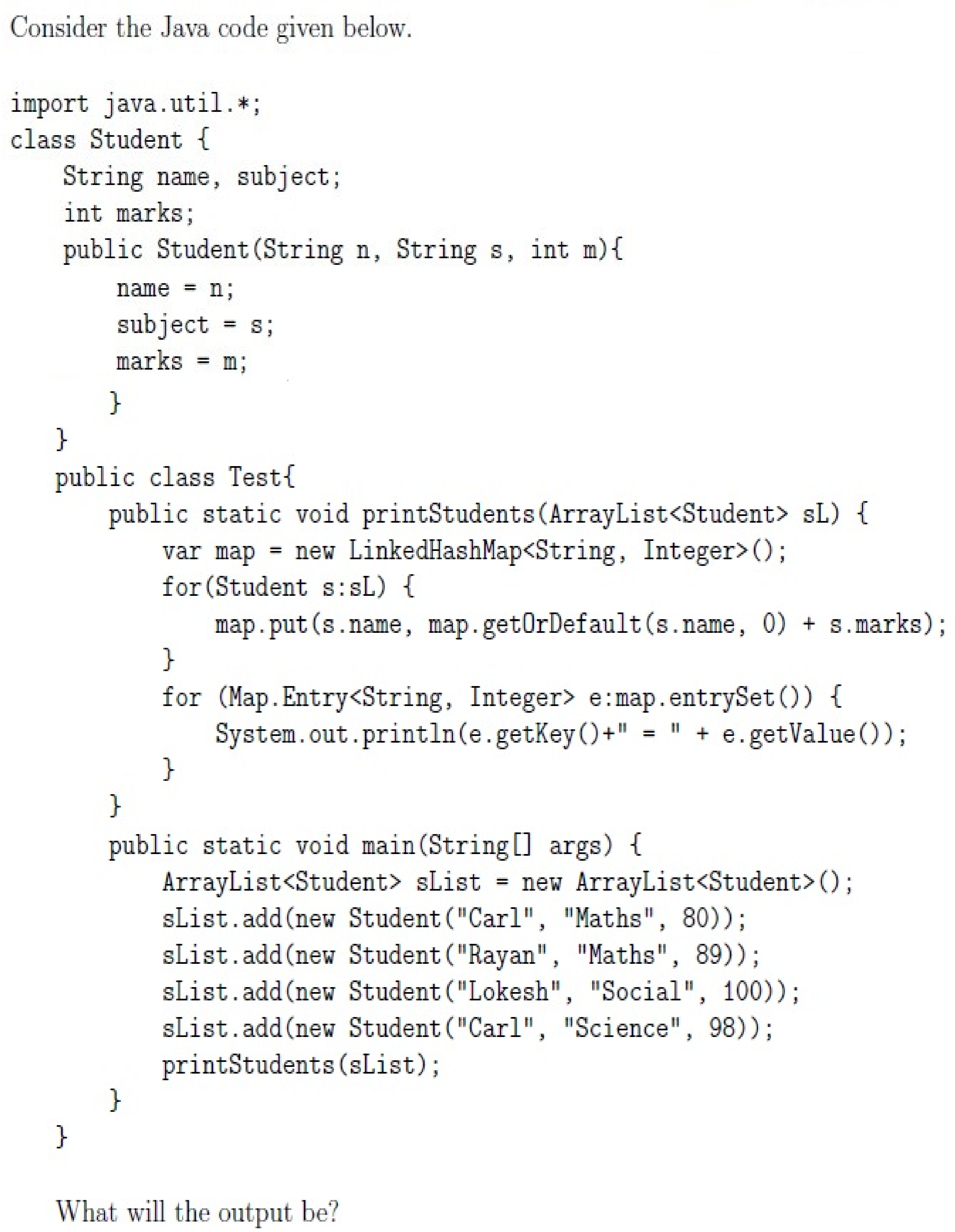
6406532306820. ✖ 

The program crashes due to the uncaught exception: `MinScoreException`

Question Number : 152 Question Id : 640653689574 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 3

Question Label : Multiple Choice Question

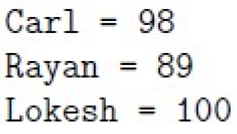


Consider the Java code given below.

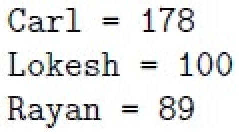
```
import java.util.*;
class Student {
    String name, subject;
    int marks;
    public Student(String n, String s, int m){
        name = n;
        subject = s;
        marks = m;
       }
   }
   public class Test{
       public static void printStudents(ArrayList<Student> sL) {
           var map = new LinkedHashMap<String, Integer>();
           for(Student s:sL) {
               map.put(s.name, map.getOrDefault(s.name, 0) + s.marks);
           }
           for (Map.Entry<String, Integer> e:map.entrySet()) {
               System.out.println(e.getKey()+" = " + e.getValue());
           }
       }
       public static void main(String[] args) {
           ArrayList<Student> sList = new ArrayList<Student>();
           sList.add(new Student("Carl", "Maths", 80));
           sList.add(new Student("Rayan", "Maths", 89));
           sList.add(new Student("Lokesh", "Social", 100));
           sList.add(new Student("Carl", "Science", 98));
           printStudents(sList);
       }
   }
```

What will the output be?

Options :

6406532306821. ✗ 

```
Carl = 98
Rayan = 89
Lokesh = 100
```

6406532306822. ✗ 

```
Carl = 178
Lokesh = 100
Rayan = 89
```

6406532306823. ✓

```
Carl = 178
Rayan = 89
Lokesh = 100
```

6406532306824. ✖

```
Carl = 80
Lokesh = 100
Rayan = 89
```

Question Number : 153 Question Id : 640653689575 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 3

Question Label : Multiple Choice Question

Consider the following code.

```
import java.util.*;
public class TreeTest {
    public static void main(String[] args) {
        ArrayList<String> list = new ArrayList<String>();
        list.add("Monkey");
        list.add("Lion");
        list.add("Goat");
        list.add("Elephant");
        HashSet<String> set1 = new HashSet<String>(list);
        TreeSet<String> set2 = new TreeSet<String>(set1);
        set2.addAll(set1);
        System.out.println(set2);
   }
}
```

Choose the correct option.

Options :

6406532306825. ✖ This program generates the output:

```
[Elephant, Goat, Lion, Monkey, Elephant, Goat, Lion, Monkey]
```

6406532306826. ✔ This program generates the output:

```
[Elephant, Goat, Lion, Monkey]
```

6406532306827. ✖ This program terminates abnormally due to UnsupportedOperationException.

6406532306828. ✖ This program generates the output:

```
[]
```

Sub-Section Number : 3

Sub-Section Id : 640653100849

Question Shuffling Allowed : Yes

Is Section Default? : null

Question Number : 154 Question Id : 640653689565 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

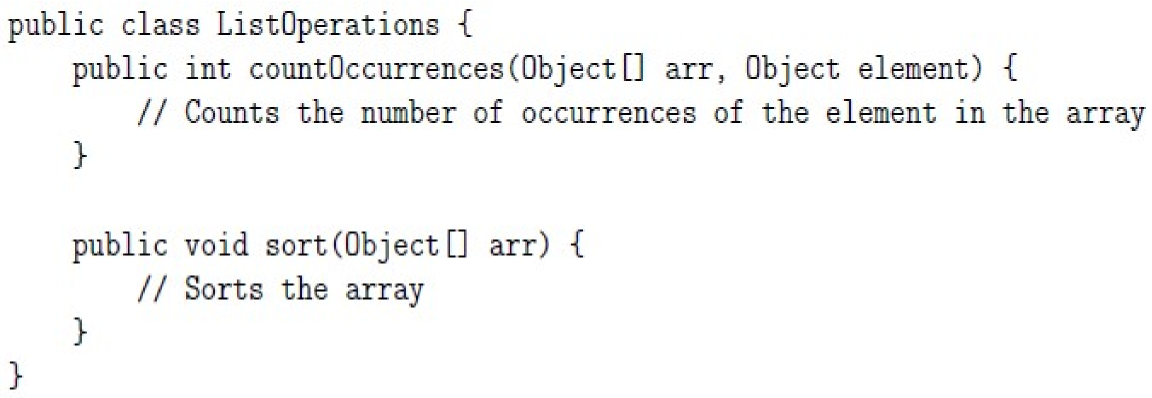
Consider the Java code given below.

```
public class ListOperations {
    public <E> int countOccurrences(E[] arr, E element) {
        // Counts the number of occurrences of the element in the array
    }

    public <E extends Comparable<E>> void sort(E[] arr) {
        // Sorts the array
    }
}
```

How does class ListOperations look after type erasure?

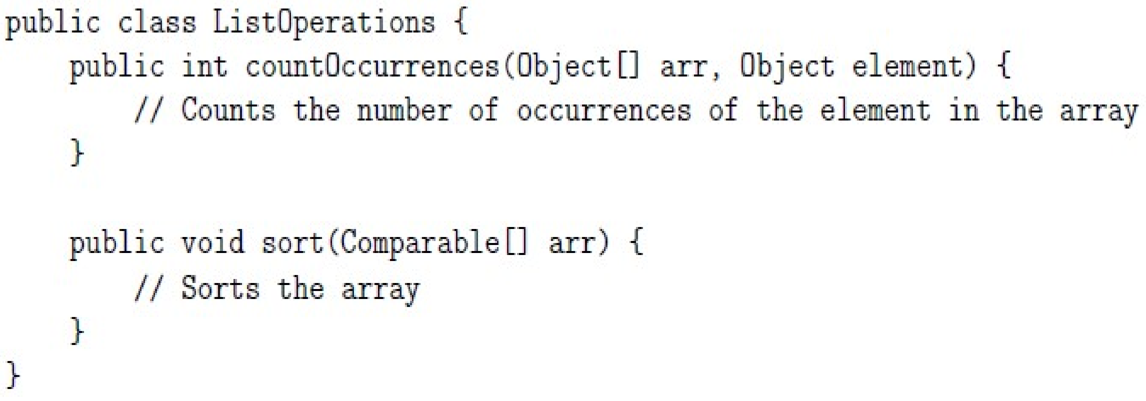
Options :

6406532306784. ✖



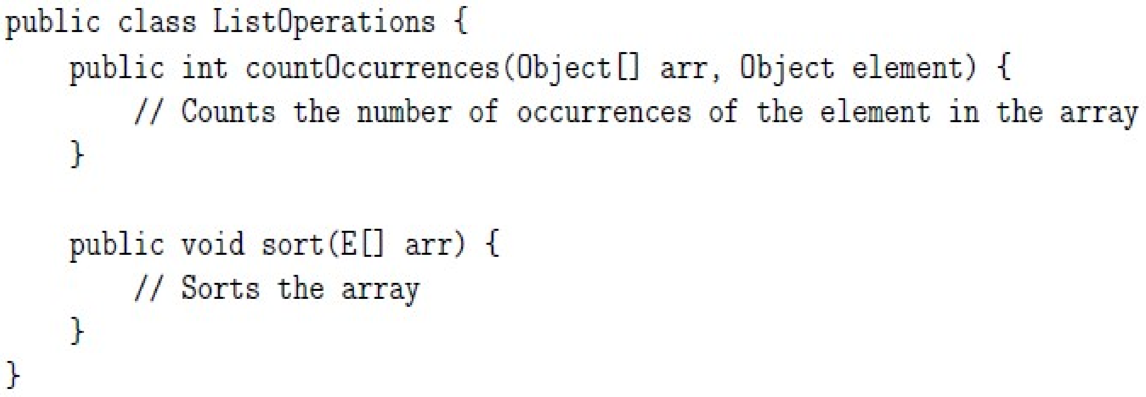
```
public class ListOperations {
    public int countOccurrences(Object[] arr, Object element) {
        // Counts the number of occurrences of the element in the array
    }

    public void sort(Object[] arr) {
        // Sorts the array
    }
}
```

6406532306785. ✓



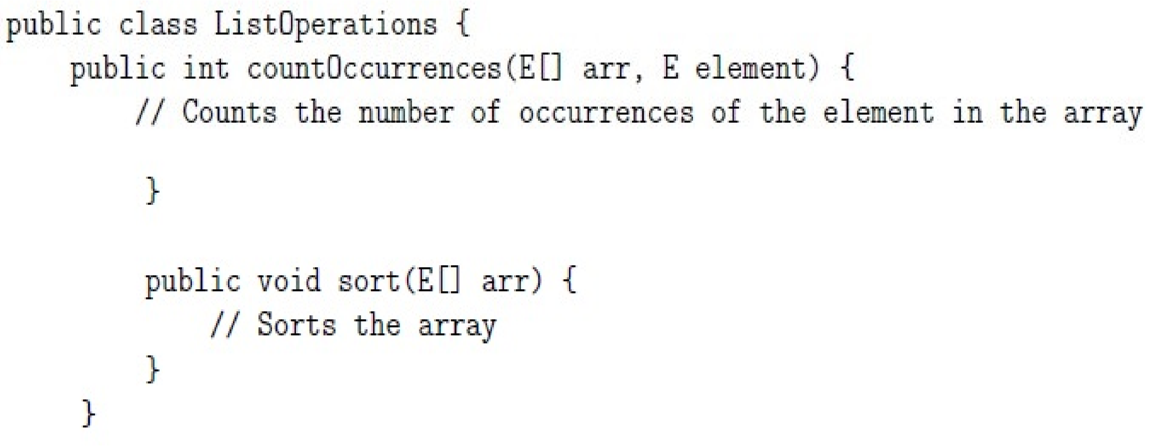
```
public class ListOperations {
    public int countOccurrences(Object[] arr, Object element) {
        // Counts the number of occurrences of the element in the array
    }

    public void sort(Comparable[] arr) {
        // Sorts the array
    }
}
```

6406532306786. ✖



```
public class ListOperations {
    public int countOccurrences(Object[] arr, Object element) {
        // Counts the number of occurrences of the element in the array
    }

    public void sort(E[] arr) {
        // Sorts the array
    }
}
```

6406532306787. ✖

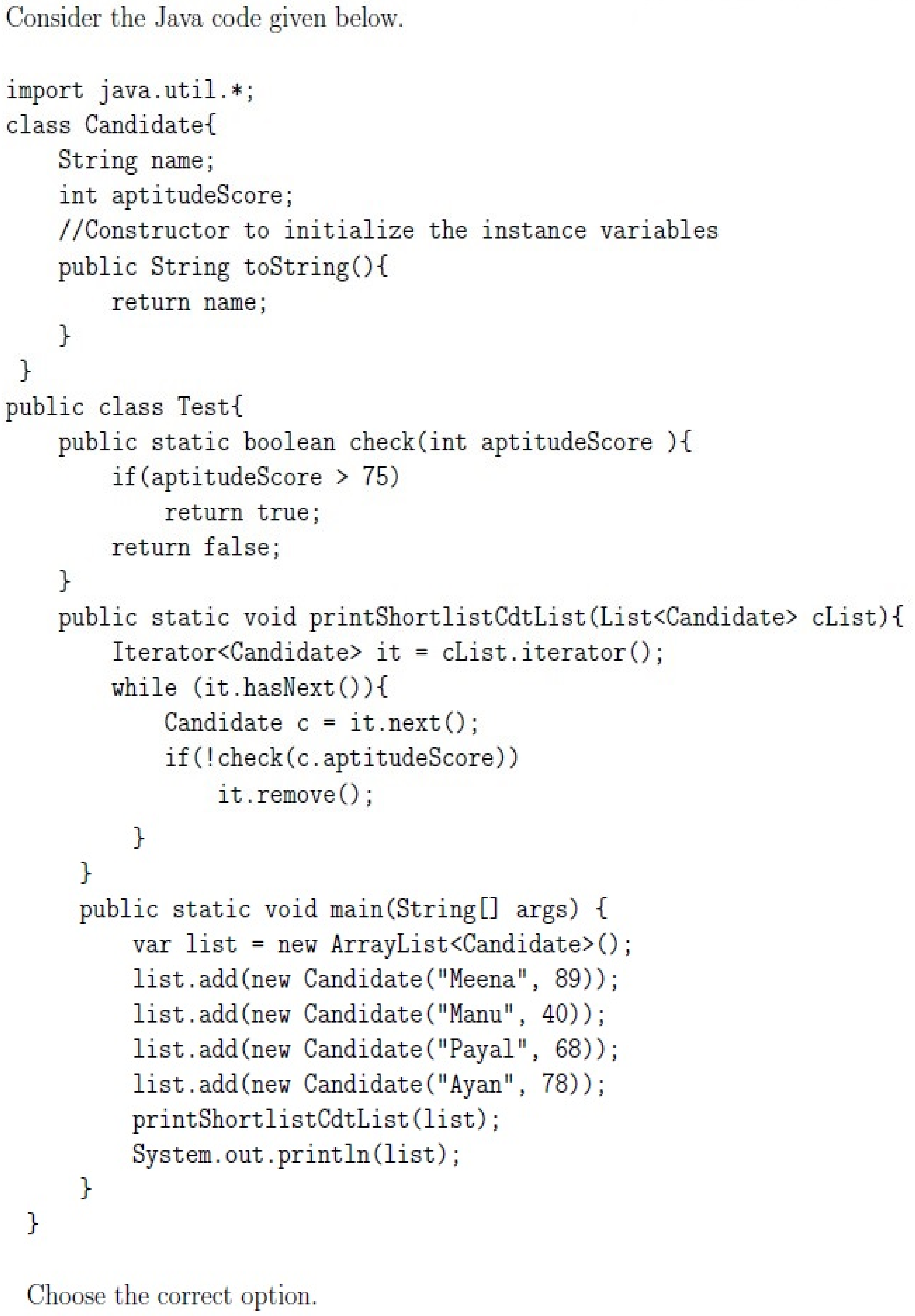


```
public class ListOperations {
    public int countOccurrences(E[] arr, E element) {
        // Counts the number of occurrences of the element in the array

        }

        public void sort(E[] arr) {
            // Sorts the array
        }
    }
```

Question Number : 155 Question Id : 640653689571 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.*;
class Candidate{
    String name;
    int aptitudeScore;
    //Constructor to initialize the instance variables
    public String toString(){
        return name;
    }
 }
public class Test{
    public static boolean check(int aptitudeScore ){
        if(aptitudeScore > 75)
            return true;
        return false;
    }
    public static void printShortlistCdtList(List<Candidate> cList){
        Iterator<Candidate> it = cList.iterator();
        while (it.hasNext()){
            Candidate c = it.next();
            if(!check(c.aptitudeScore))
                it.remove();
          }
      }
      public static void main(String[] args) {
          var list = new ArrayList<Candidate>();
          list.add(new Candidate("Meena", 89));
          list.add(new Candidate("Manu", 40));
          list.add(new Candidate("Payal", 68));
          list.add(new Candidate("Ayan", 78));
          printShortlistCdtList(list);
          System.out.println(list);
      }
  }
```

Choose the correct option.

Options :

6406532306809. ✗ This program generates the output: [Ayan, Meena]

6406532306810. ✗ This program generates the output: [Manu, Payal]

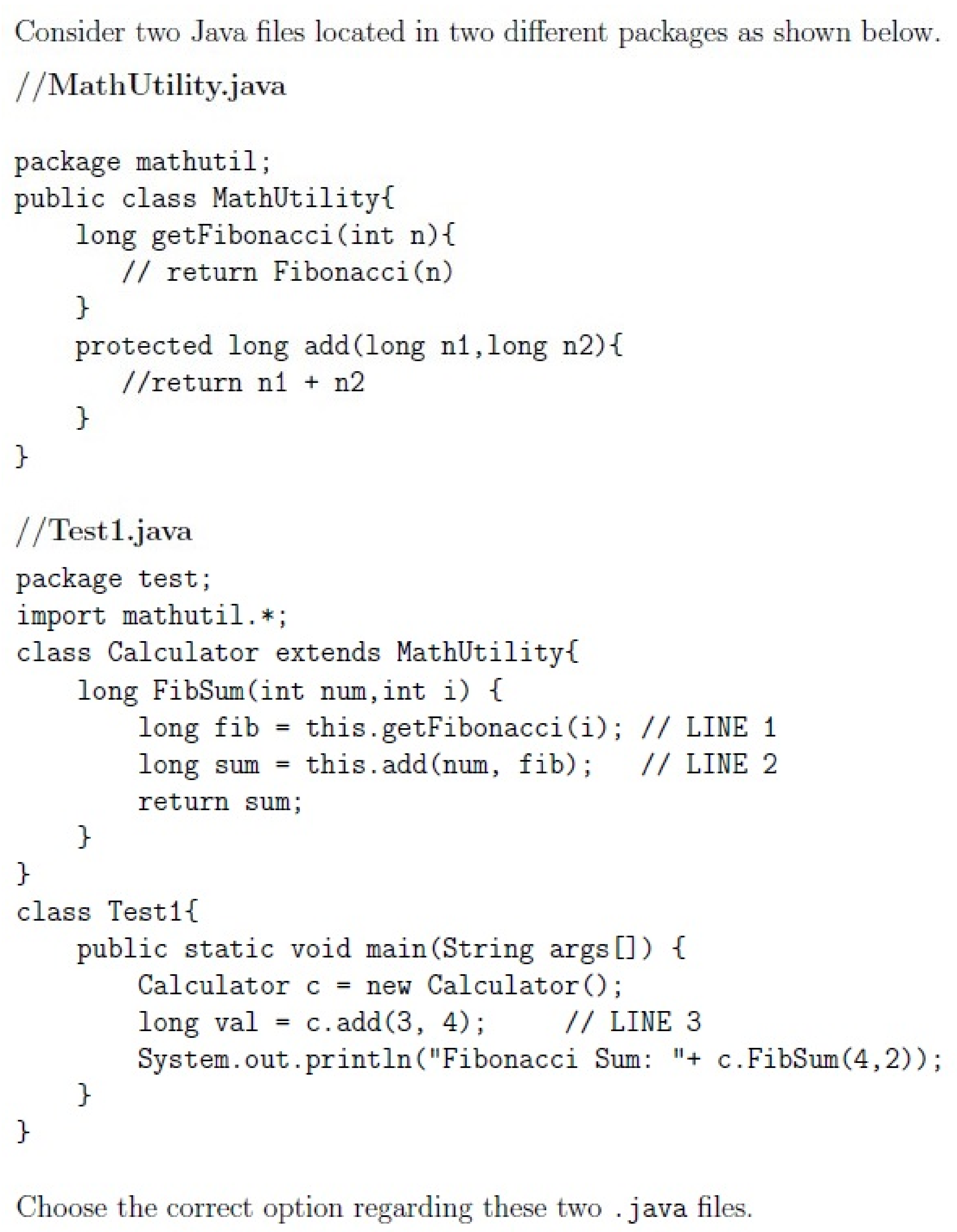
6406532306811. ✓ This program generates the output: [Meena, Ayan]

6406532306812. ✗ This program generates the output: [Ayan, Manu, Meena, Payal]

Question Number : 156 Question Id : 640653689572 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 two Java files located in two different packages as shown below.

//MathUtility.java

```
package mathutil;
public class MathUtility{
    long getFibonacci(int n){
       // return Fibonacci(n)
    }
    protected long add(long n1,long n2){
       //return n1 + n2
    }
}
```

//Test1.java

```
package test;
import mathutil.*;
class Calculator extends MathUtility{
    long FibSum(int num,int i) {
        long fib = this.getFibonacci(i); // LINE 1
        long sum = this.add(num, fib);   // LINE 2
        return sum;
    }
}
class Test1{
    public static void main(String args[]) {
        Calculator c = new Calculator();
        long val = c.add(3, 4);     // LINE 3
        System.out.println("Fibonacci Sum: "+ c.FibSum(4,2));
    }
}
```

Choose the correct option regarding these two .java files.

Options :

6406532306813. ✓ LINE 1 & LINE 3 will lead to compilation error.

6406532306814. ✗ LINE 2 & LINE 3 will lead to compilation error.

6406532306815. ✗ Only LINE 1 will lead to compilation error.

6406532306816. ✗ The code will compile successfully.

Sub-Section Number : 4

Sub-Section Id : 640653100850

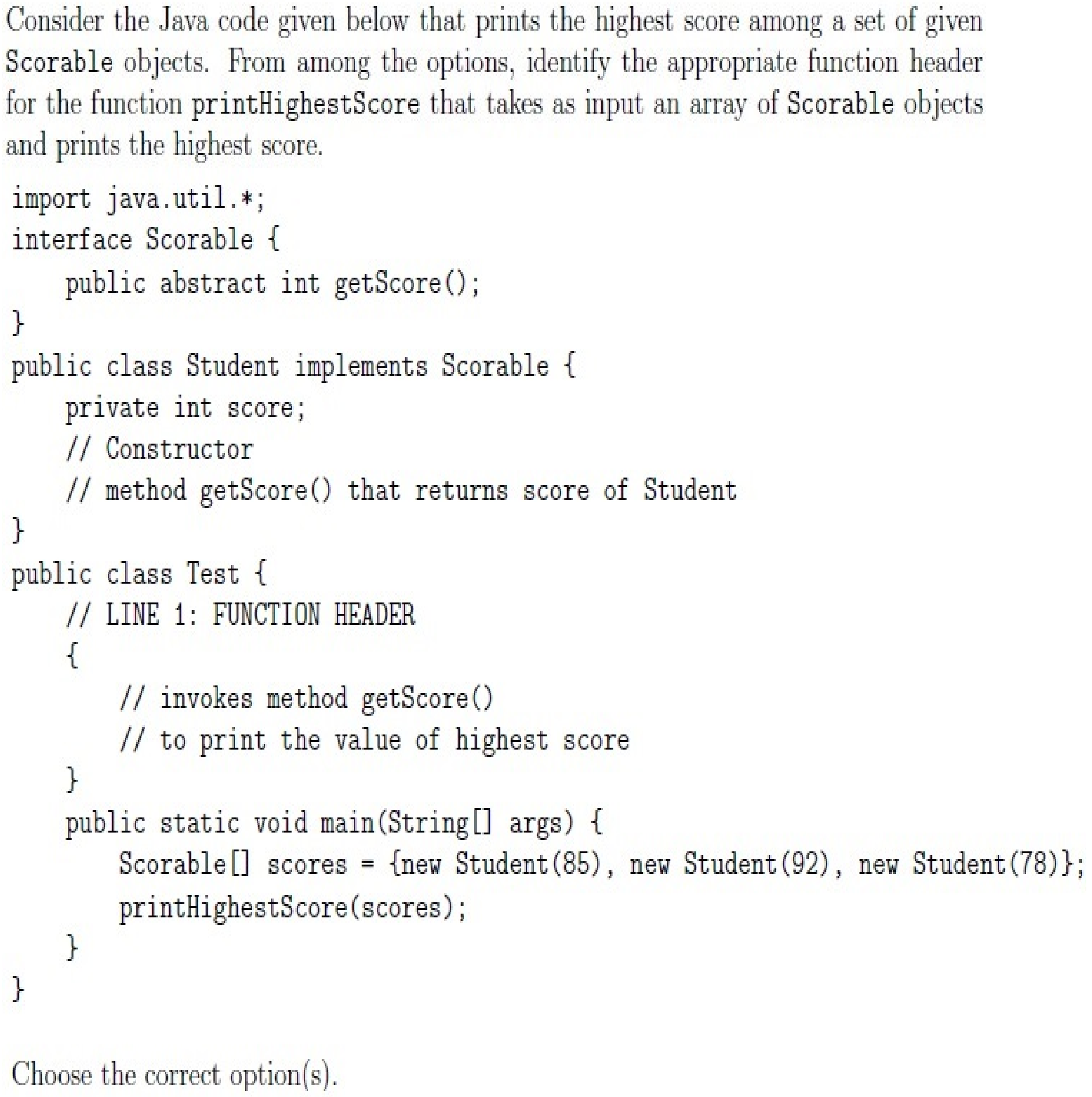
Question Shuffling Allowed : Yes

Is Section Default? : null

Question Number : 157 Question Id : 640653689563 Question Type : MSQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 3 Max. Selectable Options : 0

Question Label : Multiple Select Question

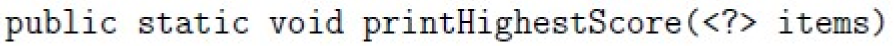


Consider the Java code given below that prints the highest score among a set of given Scorable objects. From among the options, identify the appropriate function header for the function printHighestScore that takes as input an array of Scorable objects and prints the highest score.

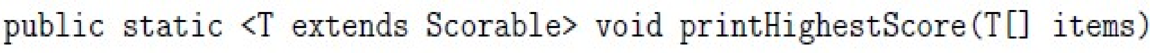
```
import java.util.*;
interface Scorable {
    public abstract int getScore();
}
public class Student implements Scorable {
    private int score;
    // Constructor
    // method getScore() that returns score of Student
}
public class Test {
    // LINE 1: FUNCTION HEADER
    {
        // invokes method getScore()
        // to print the value of highest score
    }
    public static void main(String[] args) {
        Scorable[] scores = {new Student(85), new Student(92), new Student(78)};
        printHighestScore(scores);
    }
}
```

Choose the correct option(s).

Options :

6406532306776. ✖ 

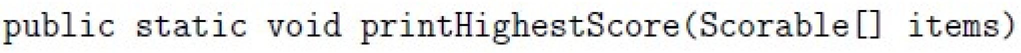
```
public static void printHighestScore(<?> items)
```

6406532306777. ✓ 

```
public static <T extends Scorable> void printHighestScore(T[] items)
```

6406532306778. ✖

```
public static <T extends Student> void printHighestScore(T[] items)
```

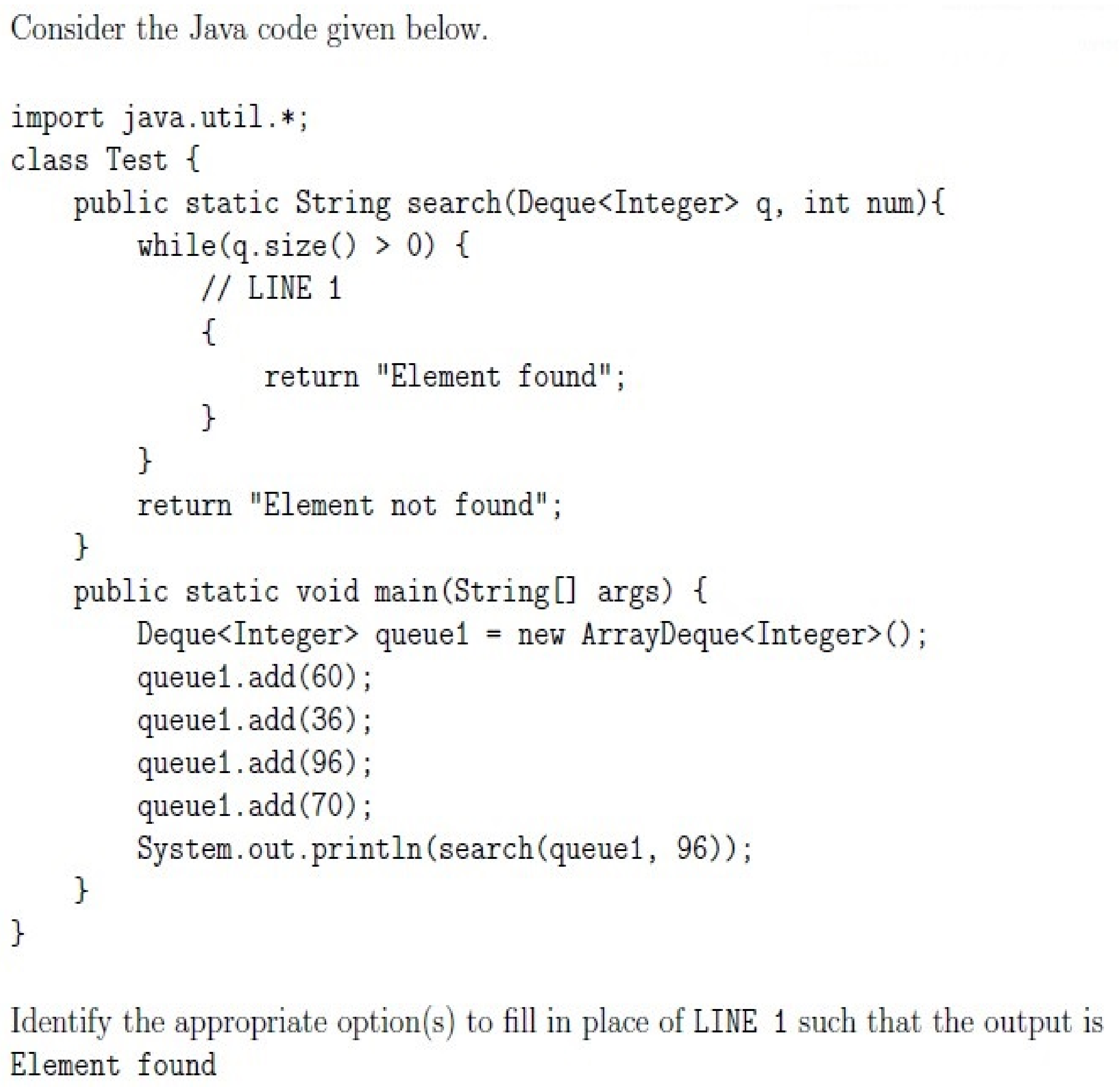
6406532306779. ✓ 

```
public static void printHighestScore(Scorable[] items)
```

Question Number : 158 Question Id : 640653689576 Question Type : MSQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 3 Max. Selectable Options : 0

Question Label : Multiple Select Question

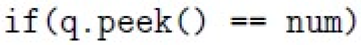


Consider the Java code given below.

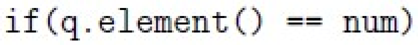
```
import java.util.*;
class Test {
    public static String search(Deque<Integer> q, int num){
        while(q.size() > 0) {
            // LINE 1
            {
                return "Element found";
            }
        }
        return "Element not found";
    }
    public static void main(String[] args) {
        Deque<Integer> queue1 = new ArrayDeque<Integer>();
        queue1.add(60);
        queue1.add(36);
        queue1.add(96);
        queue1.add(70);
        System.out.println(search(queue1, 96));
    }
}
```

Identify the appropriate option(s) to fill in place of LINE 1 such that the output is Element found

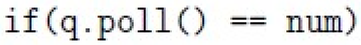
Options :

6406532306829. ✗ 

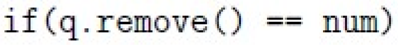
```
if(q.peek() == num)
```

6406532306830. ✗ 

```
if(q.element() == num)
```

6406532306831. ✓ 

```
if(q.poll() == num)
```

6406532306832. ✓ 

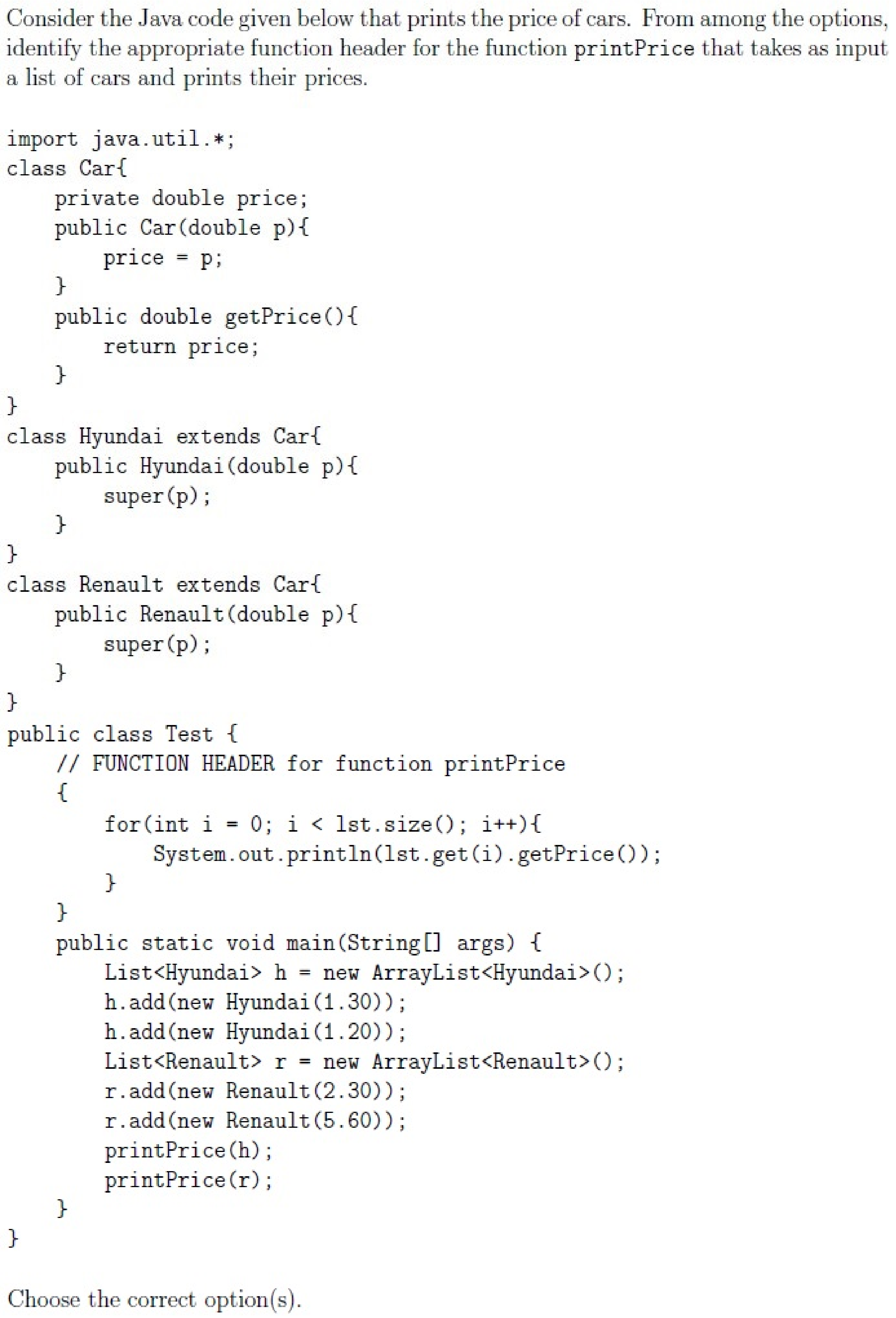
```
if(q.remove() == num)
```

Sub-Section Number :	5
Sub-Section Id :	640653100851
Question Shuffling Allowed :	Yes
Is Section Default? :	null

Question Number : 159 Question Id : 640653689566 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 price of cars. From among the options, identify the appropriate function header for the function `printPrice` that takes as input a list of cars and prints their prices.

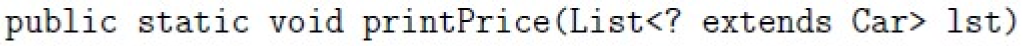
```
import java.util.*;
class Car{
    private double price;
    public Car(double p){
        price = p;
    }
    public double getPrice(){
        return price;
    }
}
class Hyundai extends Car{
    public Hyundai(double p){
        super(p);
    }
}
class Renault extends Car{
    public Renault(double p){
        super(p);
    }
}
public class Test {
    // FUNCTION HEADER for function printPrice
    {
        for(int i = 0; i < lst.size(); i++){
            System.out.println(lst.get(i).getPrice());
        }
    }
    public static void main(String[] args) {
        List<Hyundai> h = new ArrayList<Hyundai>();
        h.add(new Hyundai(1.30));
        h.add(new Hyundai(1.20));
        List<Renault> r = new ArrayList<Renault>();
        r.add(new Renault(2.30));
        r.add(new Renault(5.60));
        printPrice(h);
        printPrice(r);
    }
}
```

Choose the correct option(s).

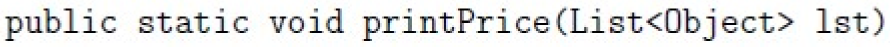
Options :

6406532306788. ✖ `public static void printPrice(List<Car> lst)`

6406532306789. ✖ `public static void printPrice(List<Renault> lst)`

6406532306790. ✓ 

```
public static void printPrice(List<? extends Car> lst)
```

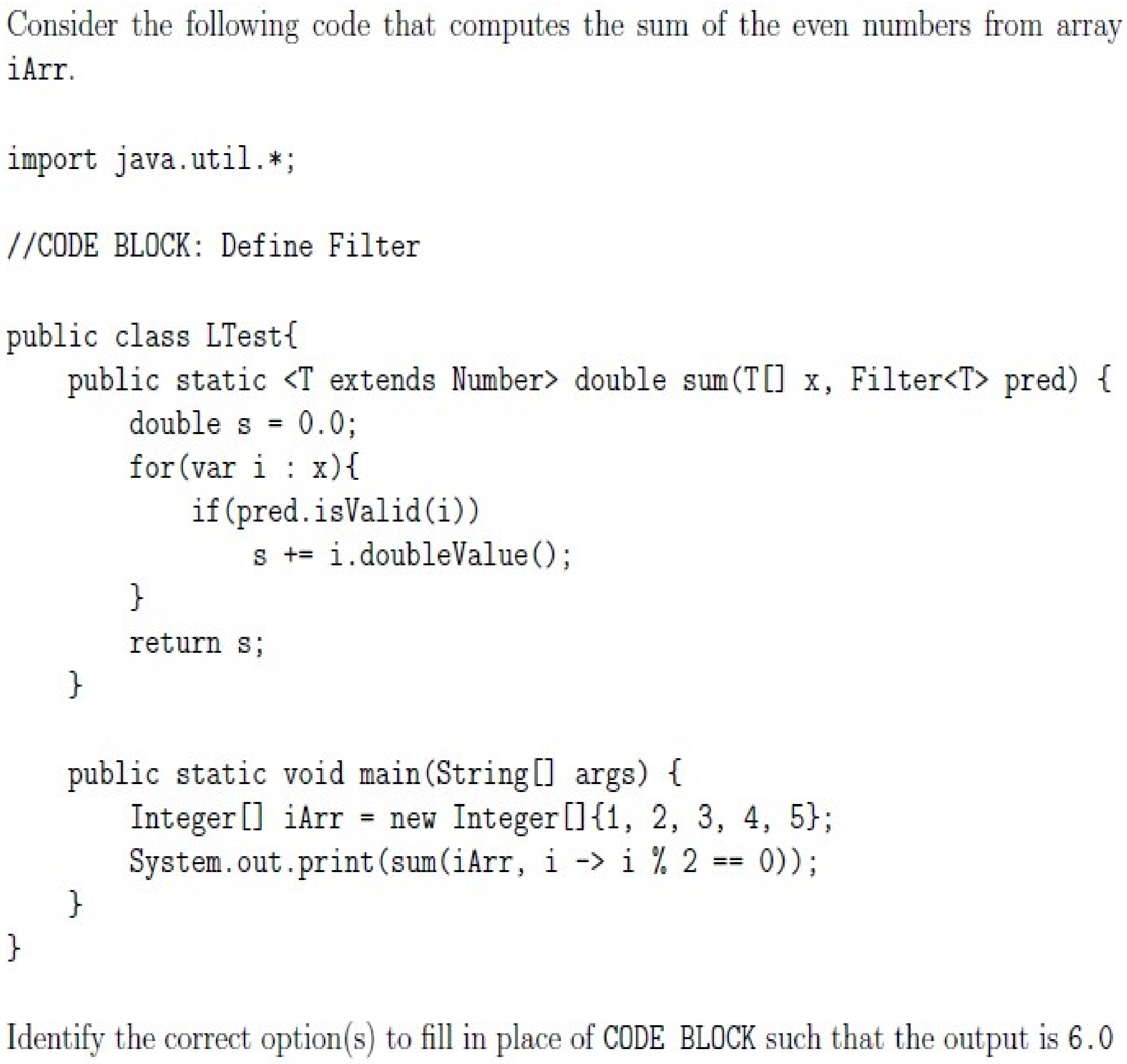
6406532306791. ✖ 

```
public static void printPrice(List<Object> lst)
```

Question Number : 160 Question Id : 640653689577 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

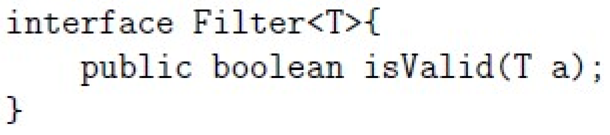


Consider the following code that computes the sum of the even numbers from array iArr.

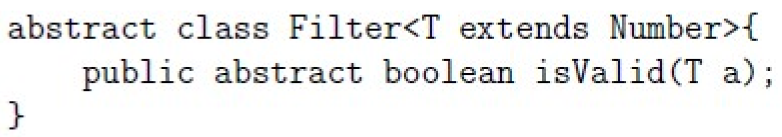
```
import java.util.*;

//CODE BLOCK: Define Filter

public class LTest{
    public static <T extends Number> double sum(T[] x, Filter<T> pred) {
        double s = 0.0;
        for(var i : x){
            if(pred.isValid(i))
                s += i.doubleValue();
        }
        return s;
    }

    public static void main(String[] args) {
        Integer[] iArr = new Integer[]{1, 2, 3, 4, 5};
        System.out.print(sum(iArr, i -> i % 2 == 0));
    }
}
```

Identify the correct option(s) to fill in place of CODE BLOCK such that the output is 6.0

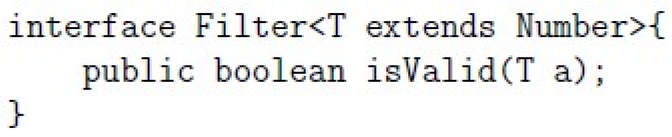
Options :

6406532306833. ✓ 

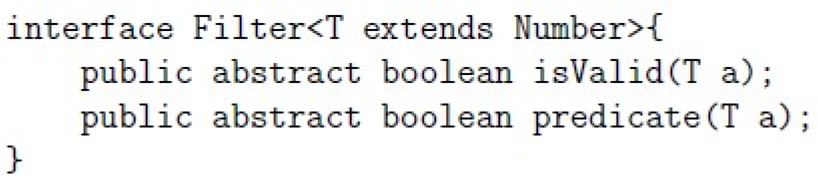
```
interface Filter<T>{
    public boolean isValid(T a);
}
```

6406532306834. ✗ 

```
abstract class Filter<T extends Number>{
    public abstract boolean isValid(T a);
}
```

6406532306835. ✓ 

```
interface Filter<T extends Number>{
    public boolean isValid(T a);
}
```

6406532306836. ✗ 

```
interface Filter<T extends Number>{
    public abstract boolean isValid(T a);
    public abstract boolean predicate(T a);
}
```

AppDev2

Section Id :	64065348509
Section Number :	11
Section type :	Online
Mandatory or Optional :	Mandatory
Number of Questions :	17
Number of Questions to be attempted :	17
Section Marks :	50
Display Number Panel :	Yes
Group All Questions :	No
Enable Mark as Answered Mark for Review and Clear Response :	Yes
Maximum Instruction Time :	0
Sub-Section Number :	1
Sub-Section Id :	640653100852
Question Shuffling Allowed :	No
Is Section Default? :	null