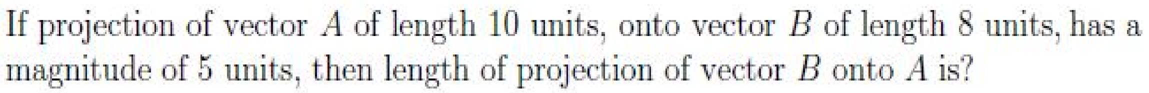
Question Label : Short Answer Question


If projection of vector A of length 10 units, onto vector B of length 8 units, has a magnitude of 5 units, then length of projection of vector B onto A is?

Response Type : Numeric

Evaluation Required For SA : Yes

Show Word Count : Yes

Answers Type : Equal

Text Areas : PlainText

Possible Answers :

4

Java

Section Id :	64065330335
Section Number :	9
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 :	64065367692
Question Shuffling Allowed :	No
Is Section Default? :	null

Question Number : 136 Question Id : 640653470062 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"

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 :

6406531562089. ✓ YES

6406531562090. ✖ NO

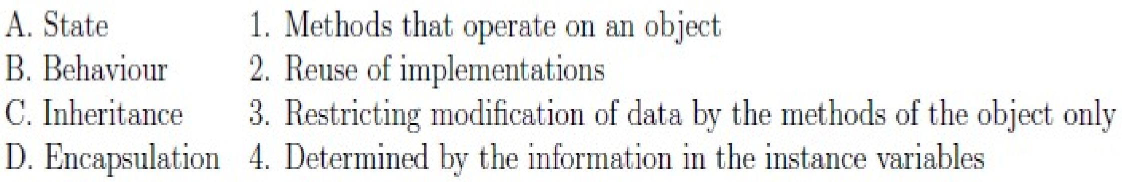
Sub-Section Number :	2
Sub-Section Id :	64065367693
Question Shuffling Allowed :	Yes
Is Section Default? :	null

Question Number : 137 Question Id : 640653470064 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

Match the following terms with their descriptions.



A. State	1. Methods that operate on an object
B. Behaviour	2. Reuse of implementations
C. Inheritance	3. Restricting modification of data by the methods of the object only
D. Encapsulation	4. Determined by the information in the instance variables

Options :

6406531562095. ✓ A-4, B-1, C-2, D-3

6406531562096. ✗ A-1, B-4, C-3, D-2

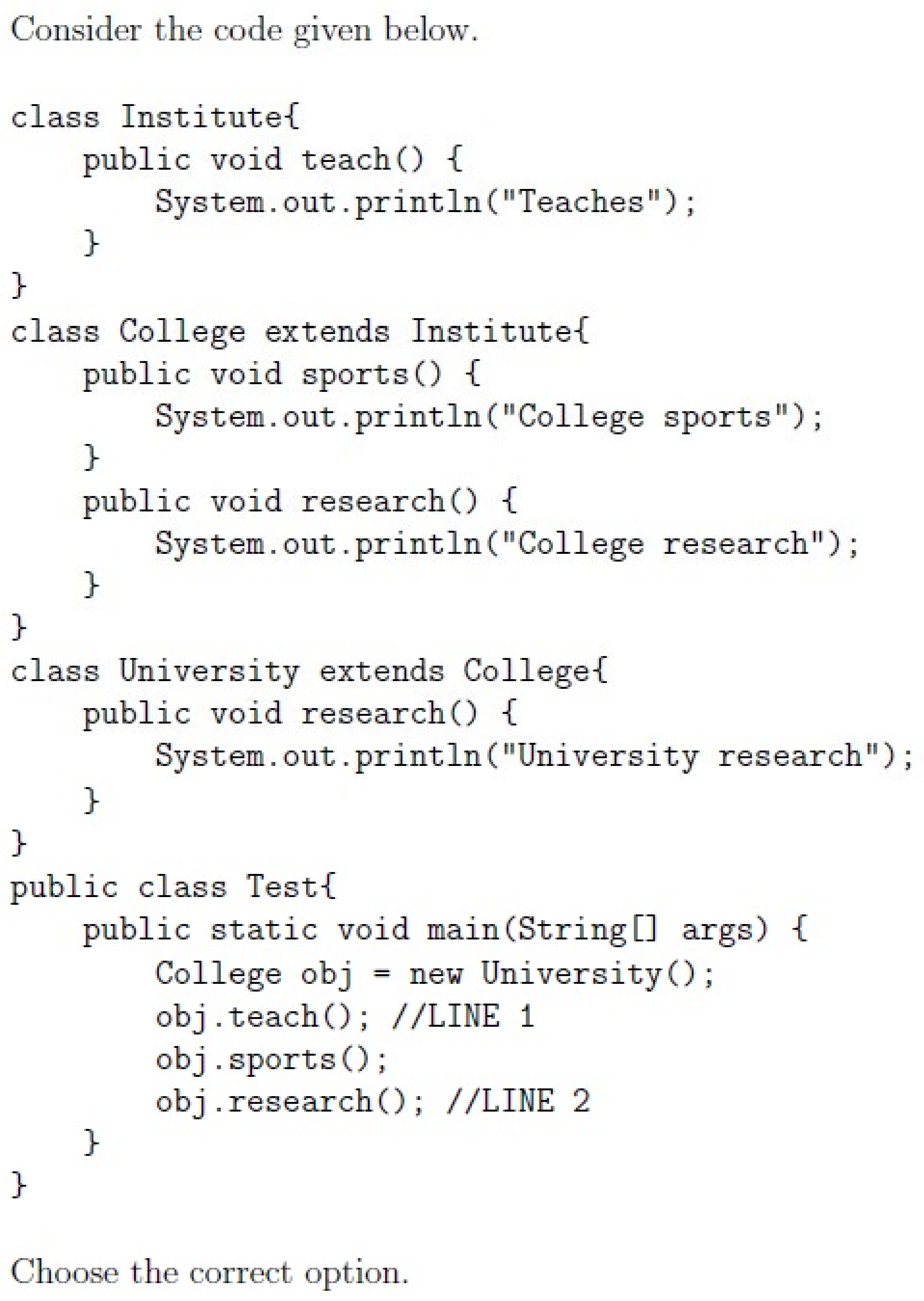
6406531562097. ✗ A-4, B-2, C-1, D-3

6406531562098. ✗ A-2, B-1, C-4, D-3

Question Number : 138 Question Id : 640653470066 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

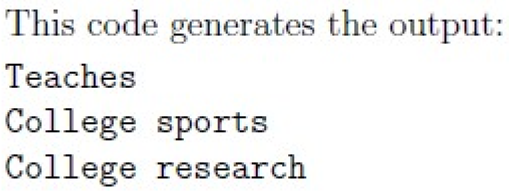


Consider the code given below.

```
class Institute{
    public void teach() {
        System.out.println("Teaches");
    }
}
class College extends Institute{
    public void sports() {
        System.out.println("College sports");
    }
    public void research() {
        System.out.println("College research");
    }
}
class University extends College{
    public void research() {
        System.out.println("University research");
    }
}
public class Test{
    public static void main(String[] args) {
        College obj = new University();
        obj.teach(); //LINE 1
        obj.sports();
        obj.research(); //LINE 2
    }
}
```

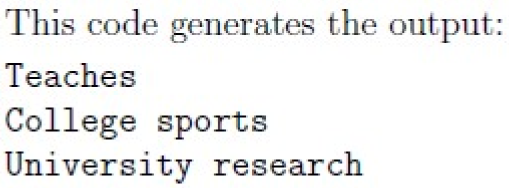
Choose the correct option.

Options :

6406531562103. ✖ 

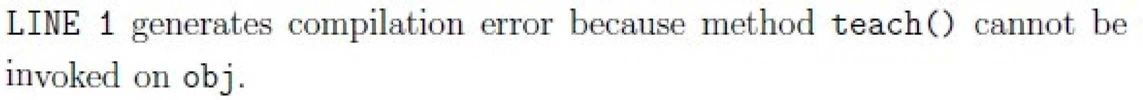
This code generates the output:

```
Teaches
College sports
College research
```

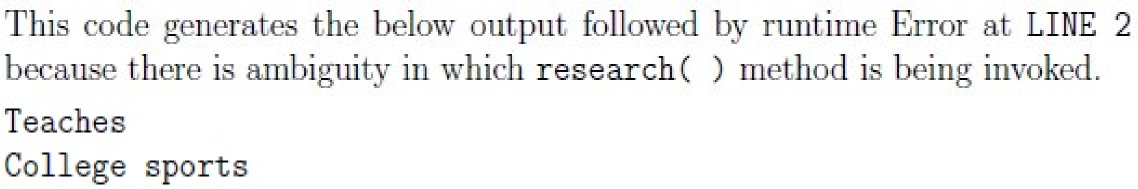
6406531562104. ✓ 

This code generates the output:

```
Teaches
College sports
University research
```

6406531562105. ✖ 

LINE 1 generates compilation error because method `teach()` cannot be invoked on `obj`.

6406531562106. ✖ 

This code generates the below output followed by runtime Error at LINE 2 because there is ambiguity in which `research( )` method is being invoked.

```
Teaches
College sports
```

Question Number : 139 Question Id : 640653470068 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 code given below.

```
class Employee{
    public final void bonus(){
        System.out.println("Employee bonus");
    }
}
class TeamManager extends Employee{
    public void bonus(){
         System.out.println("Team manager bonus");
    }
}
class Manager extends TeamManager{
}
public class  Example{
    public static void main(String[] args){
        Manager m = new Manager();
        m.bonus();
    }
}
```

Choose the correct option regarding the given code.

Options :

6406531562111. ✖ This code generates a compile time error because the method `bonus()` is not defined in the class `Manager`.

6406531562112. ✔ This code generates a compile time error because the method `bonus()` cannot be overridden.

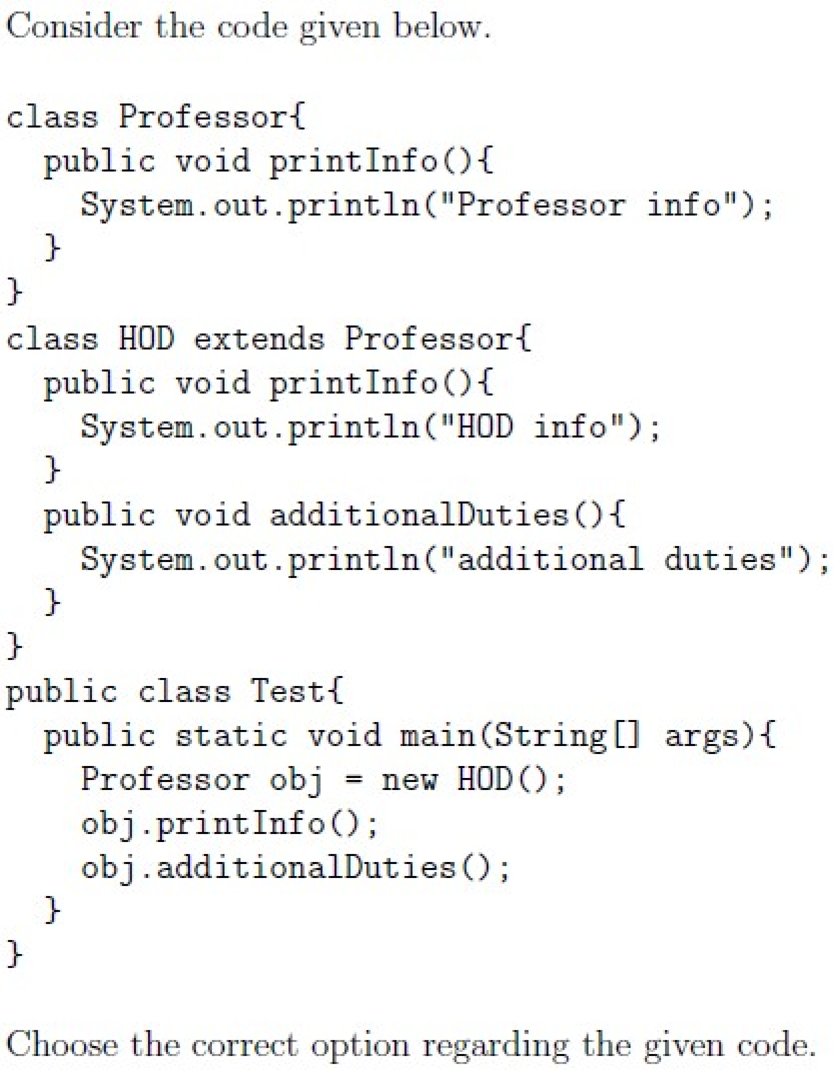
6406531562113. ✖ This code generates output:
`Employee bonus`

6406531562114. ✖ This code generates output:
`Team manager bonus`

Question Number : 140 Question Id : 640653470069 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



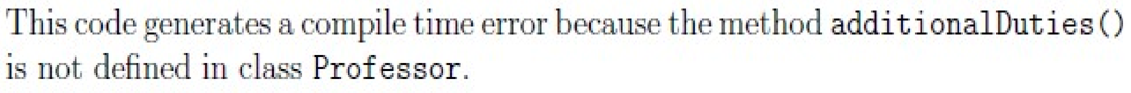
Consider the code given below.

```
class Professor{
  public void printInfo(){
    System.out.println("Professor info");
  }
}
class HOD extends Professor{
  public void printInfo(){
    System.out.println("HOD info");
  }
  public void additionalDuties(){
    System.out.println("additional duties");
  }
}
public class Test{
  public static void main(String[] args){
    Professor obj = new HOD();
    obj.printInfo();
    obj.additionalDuties();
  }
}
```

Choose the correct option regarding the given code.

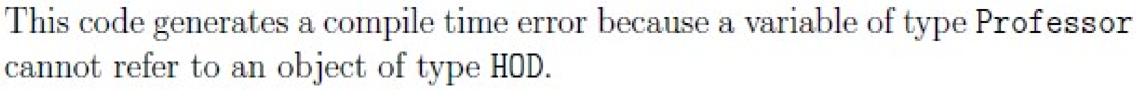
Options :

6406531562115. ✓



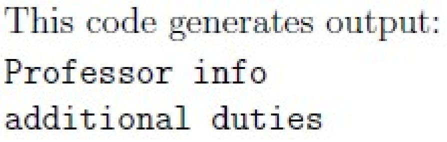
This code generates a compile time error because the method `additionalDuties()` is not defined in class `Professor`.

6406531562116. ✖



This code generates a compile time error because a variable of type `Professor` cannot refer to an object of type `HOD`.

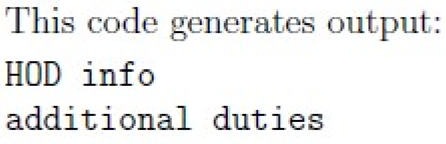
6406531562117. ✖



This code generates output:

```
Professor info
additional duties
```

6406531562118. ✖



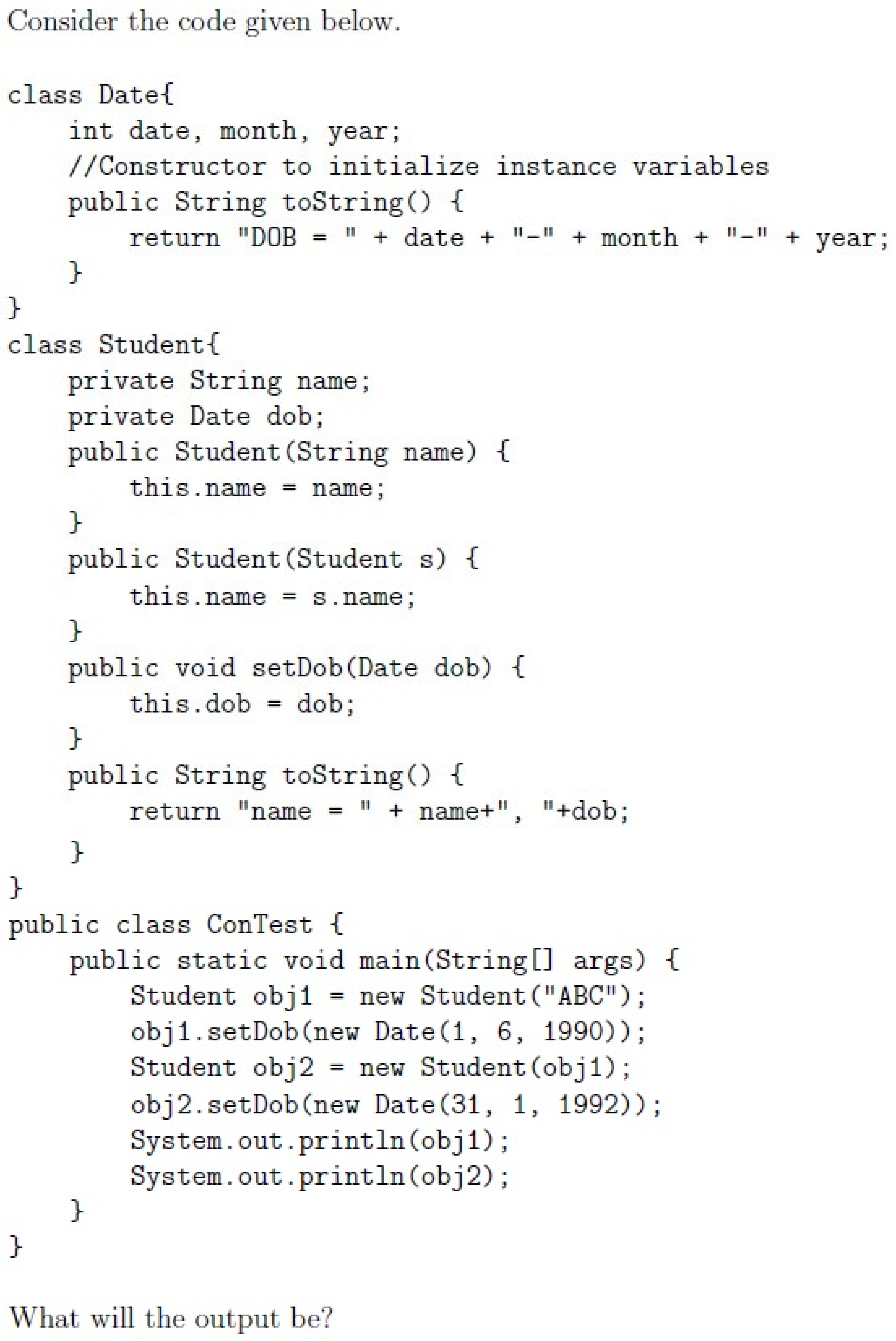
This code generates output:

```
HOD info
additional duties
```

Question Number : 141 Question Id : 640653470071 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



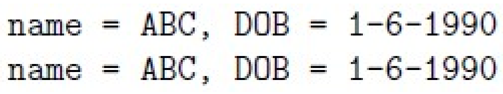
Consider the code given below.

```
class Date{
    int date, month, year;
    //Constructor to initialize instance variables
    public String toString() {
        return "DOB = " + date + "-" + month + "-" + year;
    }
}
class Student{
    private String name;
    private Date dob;
    public Student(String name) {
        this.name = name;
    }
    public Student(Student s) {
        this.name = s.name;
    }
    public void setDob(Date dob) {
        this.dob = dob;
    }
    public String toString() {
        return "name = " + name+", "+dob;
    }
}
public class ConTest {
    public static void main(String[] args) {
        Student obj1 = new Student("ABC");
        obj1.setDob(new Date(1, 6, 1990));
        Student obj2 = new Student(obj1);
        obj2.setDob(new Date(31, 1, 1992));
        System.out.println(obj1);
        System.out.println(obj2);
    }
}
```

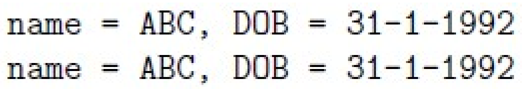
What will the output be?

Options :

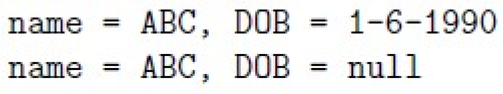
6406531562123. ✖



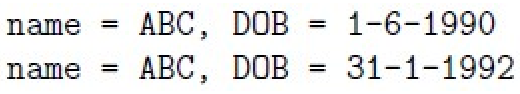
```
name = ABC, DOB = 1-6-1990
name = ABC, DOB = 1-6-1990
```

6406531562124. ✖ 

```
name = ABC, DOB = 31-1-1992
name = ABC, DOB = 31-1-1992
```

6406531562125. ✖ 

```
name = ABC, DOB = 1-6-1990
name = ABC, DOB = null
```

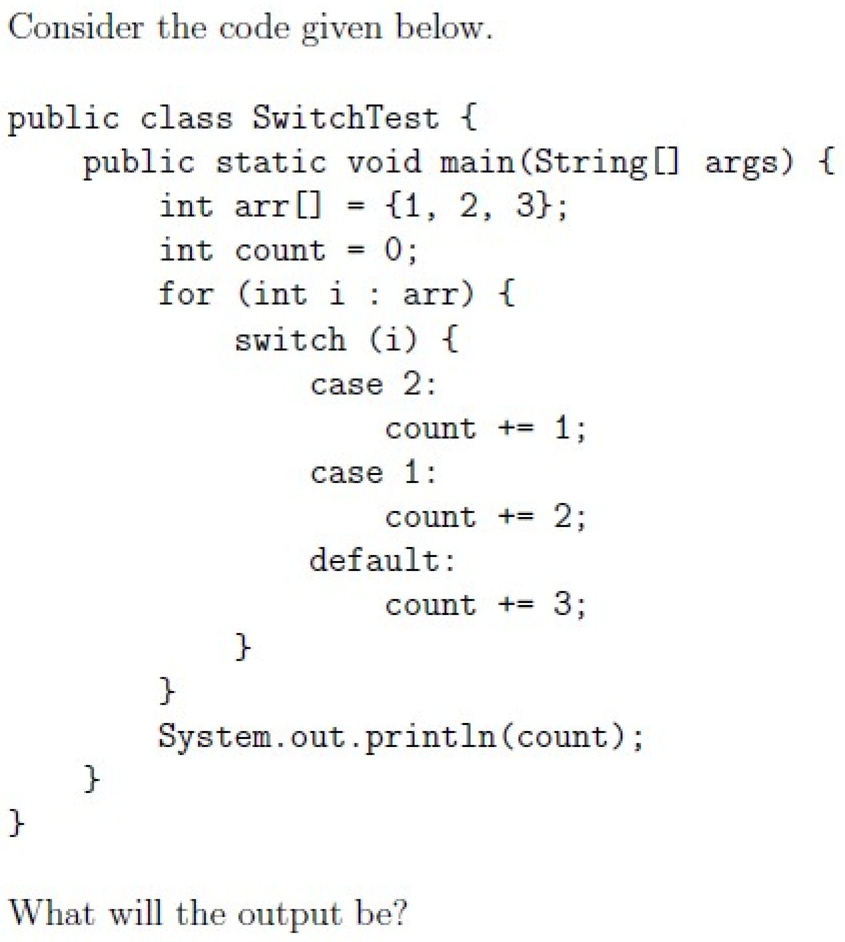
6406531562126. ✔ 

```
name = ABC, DOB = 1-6-1990
name = ABC, DOB = 31-1-1992
```

Question Number : 142 Question Id : 640653470072 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 code given below.

```
public class SwitchTest {
    public static void main(String[] args) {
        int arr[] = {1, 2, 3};
        int count = 0;
        for (int i : arr) {
            switch (i) {
                case 2:
                    count += 1;
                case 1:
                    count += 2;
                default:
                    count += 3;
            }
        }
        System.out.println(count);
    }
}
```

What will the output be?

Options :

6406531562127. ✖ 0

6406531562128. ✖ 3

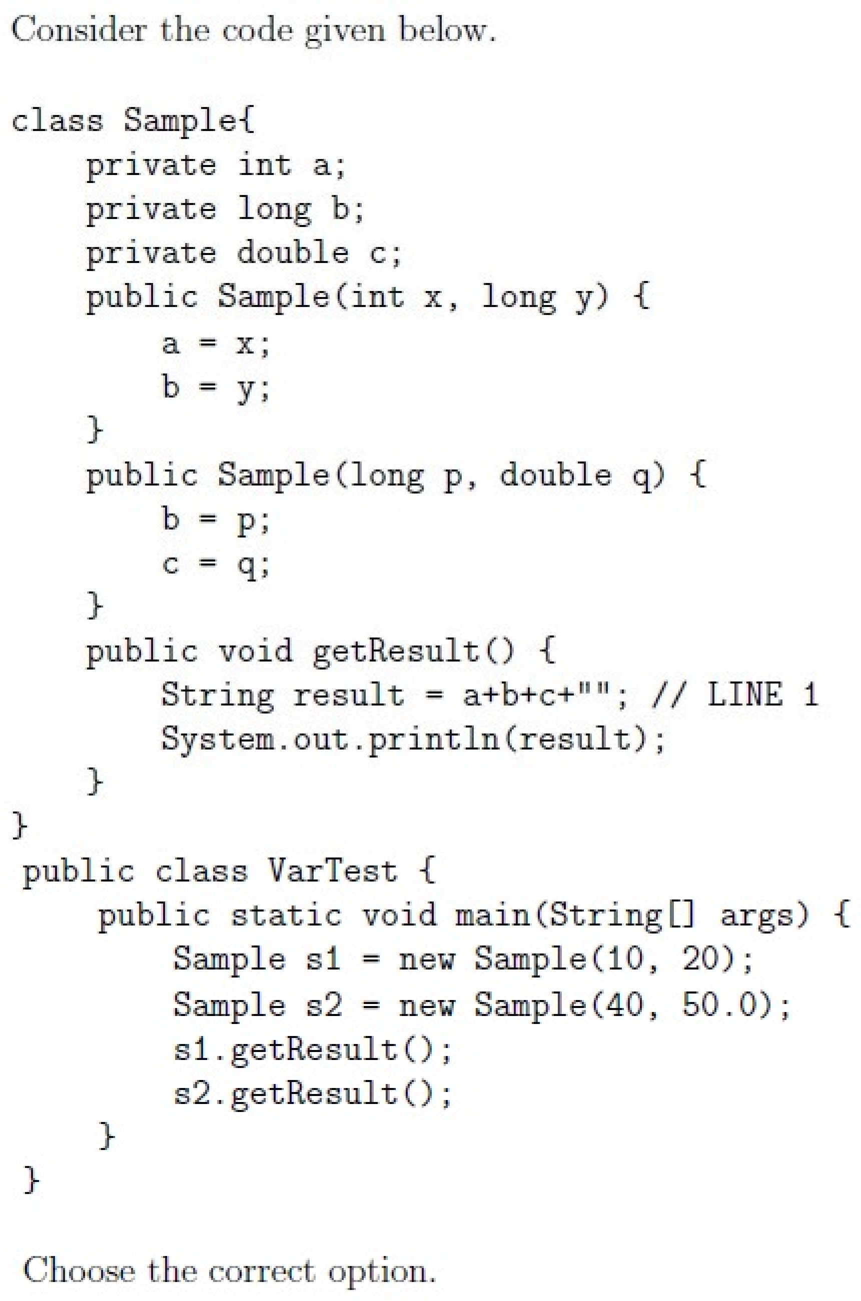
6406531562129. ✓ 14

6406531562130. ✖ 6

Question Number : 143 Question Id : 640653470073 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 code given below.

```
class Sample{
    private int a;
    private long b;
    private double c;
    public Sample(int x, long y) {
        a = x;
        b = y;
    }
    public Sample(long p, double q) {
        b = p;
        c = q;
    }
    public void getResult() {
        String result = a+b+c+""; // LINE 1
        System.out.println(result);
    }
}
 public class VarTest {
     public static void main(String[] args) {
         Sample s1 = new Sample(10, 20);
         Sample s2 = new Sample(40, 50.0);
         s1.getResult();
         s2.getResult();
     }
 }
```

Choose the correct option.

Options :

6406531562131. ✖ Compilation error at LINE 1

6406531562132. ✔ This program generates the output:

```
30.0
90.0
```

6406531562133. ✖ This program generates the output:

```
30.0
100.0
```

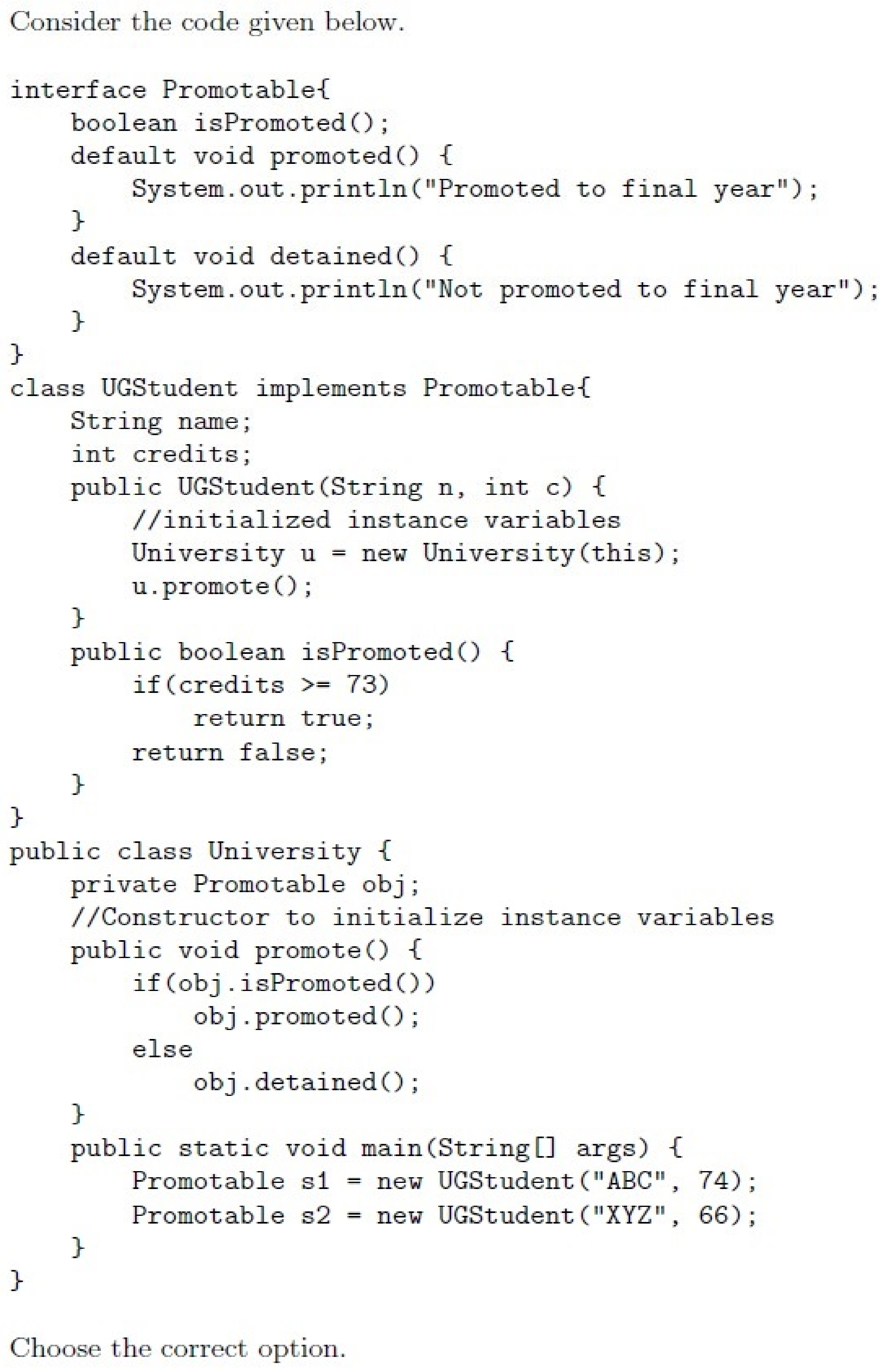
6406531562134. ✖ This program generates the output:

```
100.0
100.0
```

Question Number : 144 Question Id : 640653470077 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

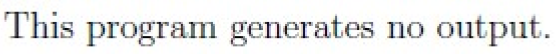


Consider the code given below.

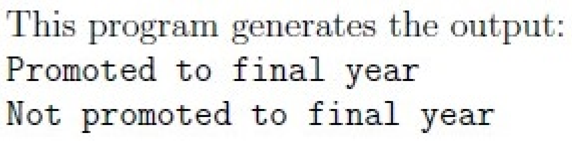
```
interface Promotable{
    boolean isPromoted();
    default void promoted() {
        System.out.println("Promoted to final year");
    }
    default void detained() {
        System.out.println("Not promoted to final year");
    }
}
class UGStudent implements Promotable{
    String name;
    int credits;
    public UGStudent(String n, int c) {
        //initialized instance variables
        University u = new University(this);
        u.promote();
    }
    public boolean isPromoted() {
        if(credits >= 73)
            return true;
        return false;
    }
}
public class University {
    private Promotable obj;
    //Constructor to initialize instance variables
    public void promote() {
        if(obj.isPromoted())
            obj.promoted();
        else
            obj.detained();
    }
    public static void main(String[] args) {
        Promotable s1 = new UGStudent("ABC", 74);
        Promotable s2 = new UGStudent("XYZ", 66);
    }
}
```

Choose the correct option.

Options :

6406531562147. ✖ 

This program generates no output.

6406531562148. ✓ 

This program generates the output:

```
Promoted to final year
Not promoted to final year
```

6406531562149. ✖ Program generates compilation error because methods `promoted()` and `detained()` should be overridden by the class `UGStudent`.

6406531562150. ✖ This program generates the output:

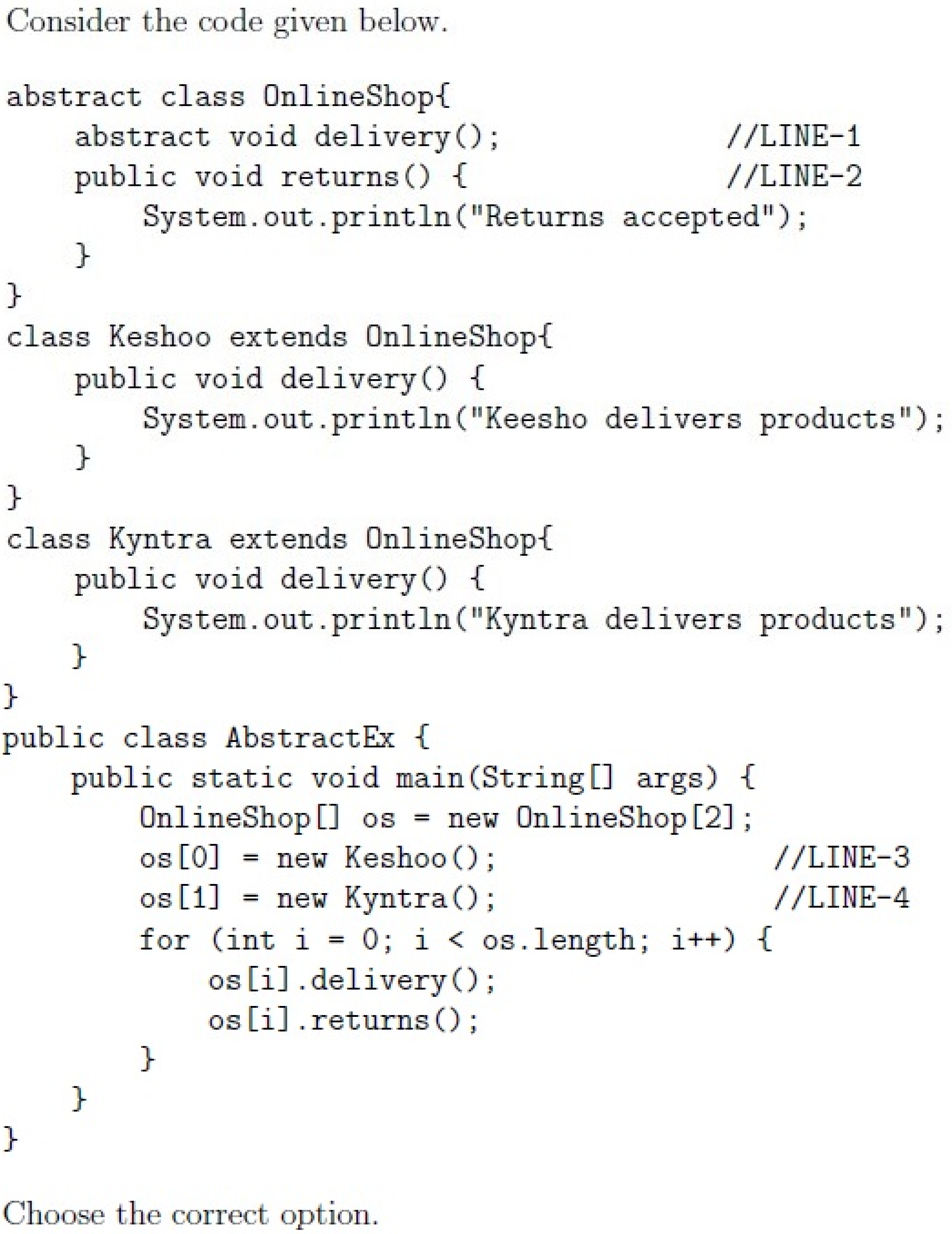
```
Not promoted to final year
Promoted to final year
```

Sub-Section Number :	3
Sub-Section Id :	64065367694
Question Shuffling Allowed :	Yes
Is Section Default? :	null

Question Number : 145 Question Id : 640653470074 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

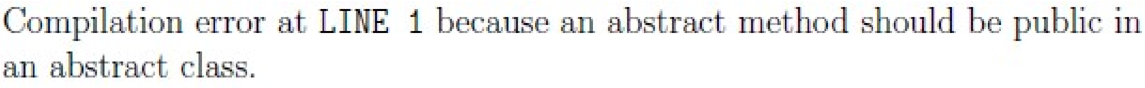


Consider the code given below.

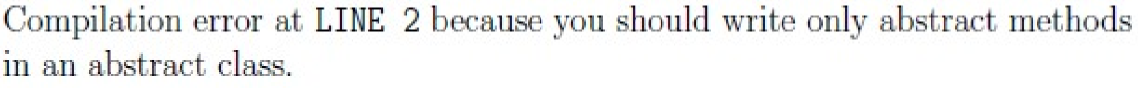
```
abstract class OnlineShop{
    abstract void delivery();                //LINE-1
    public void returns() {                  //LINE-2
        System.out.println("Returns accepted");
    }
}
class Keshoo extends OnlineShop{
    public void delivery() {
        System.out.println("Keesho delivers products");
    }
}
class Kyntra extends OnlineShop{
    public void delivery() {
        System.out.println("Kyntra delivers products");
    }
}
public class AbstractEx {
    public static void main(String[] args) {
        OnlineShop[] os = new OnlineShop[2];
        os[0] = new Keshoo();                   //LINE-3
        os[1] = new Kyntra();                   //LINE-4
        for (int i = 0; i < os.length; i++) {
            os[i].delivery();
            os[i].returns();
        }
    }
}
```

Choose the correct option.

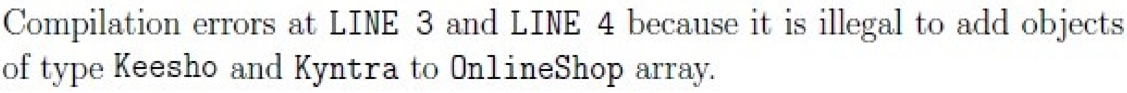
Options :

6406531562135. ✖ 

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

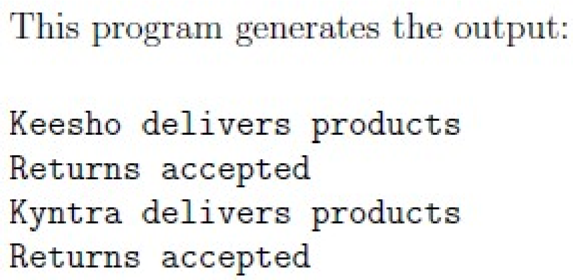
6406531562136. ✖ 

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

6406531562137. ✖ 

Compilation errors at `LINE 3` and `LINE 4` because it is illegal to add objects of type `Keesho` and `Kyntra` to `OnlineShop` array.

6406531562138. ✓



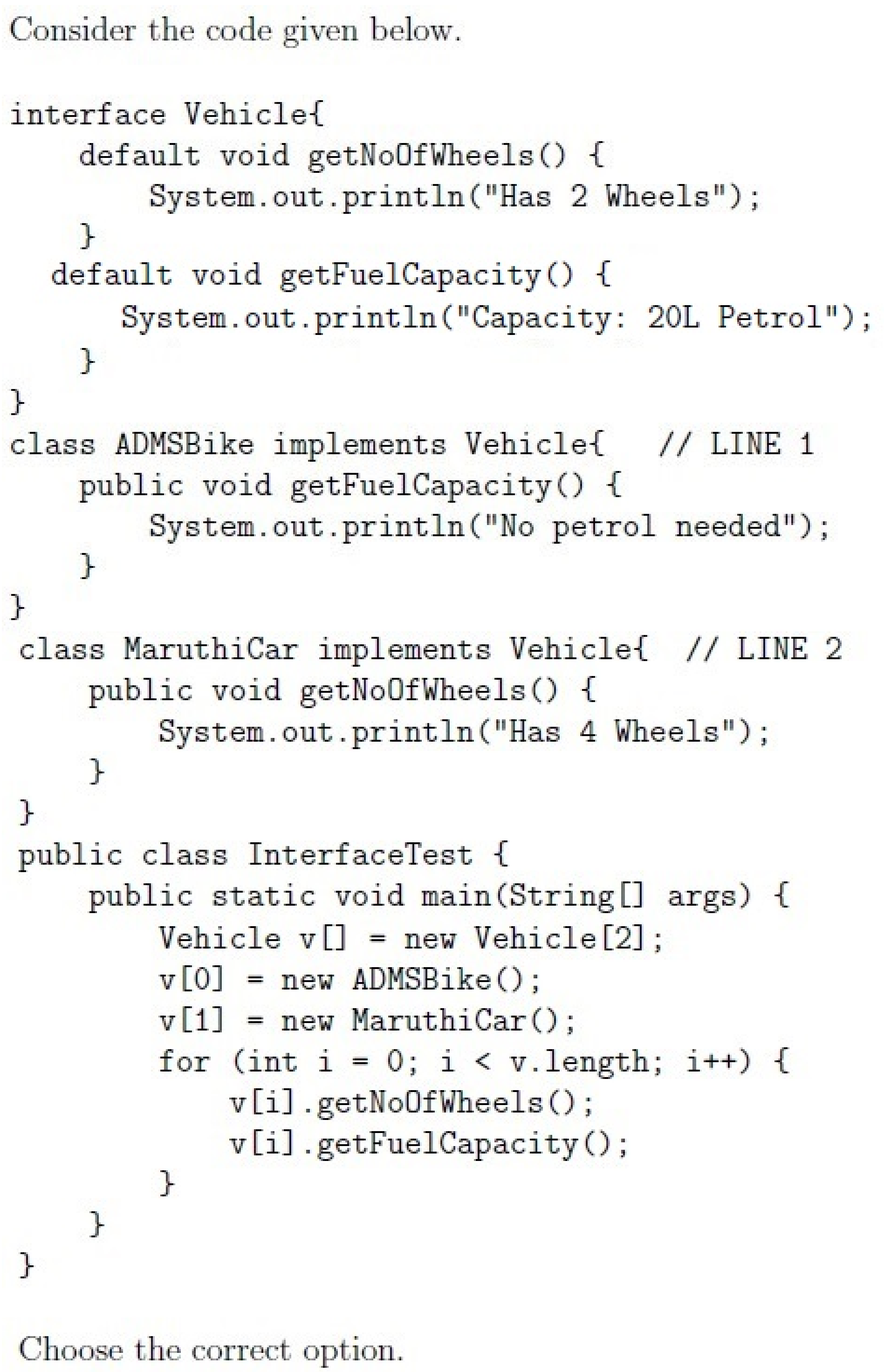
This program generates the output:

```
Keesho delivers products
Returns accepted
Kyntra delivers products
Returns accepted
```

Question Number : 146 Question Id : 640653470075 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

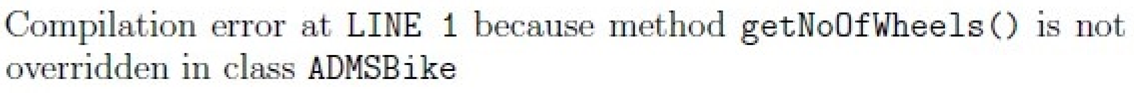


Consider the code given below.

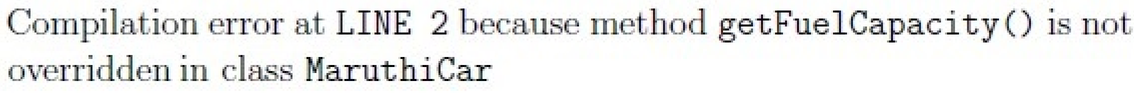
```
interface Vehicle{
    default void getNoOfWheels() {
        System.out.println("Has 2 Wheels");
    }
  default void getFuelCapacity() {
      System.out.println("Capacity: 20L Petrol");
    }
}
class ADMSBike implements Vehicle{   // LINE 1
    public void getFuelCapacity() {
        System.out.println("No petrol needed");
    }
}
 class MaruthiCar implements Vehicle{  // LINE 2
     public void getNoOfWheels() {
         System.out.println("Has 4 Wheels");
     }
 }
 public class InterfaceTest {
     public static void main(String[] args) {
         Vehicle v[] = new Vehicle[2];
         v[0] = new ADMSBike();
         v[1] = new MaruthiCar();
         for (int i = 0; i < v.length; i++) {
             v[i].getNoOfWheels();
             v[i].getFuelCapacity();
         }
     }
 }
```

Choose the correct option.

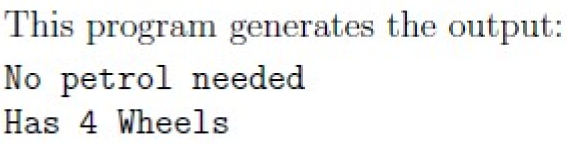
Options :

6406531562139. ✖ 

Compilation error at LINE 1 because method getNoOfWheels() is not overridden in class ADMSBike

6406531562140. ✖ 

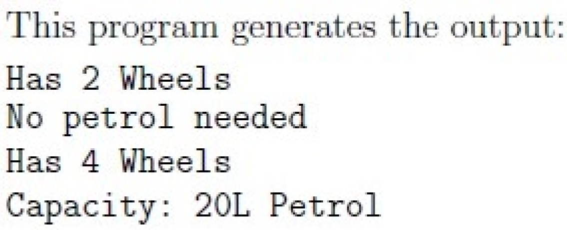
Compilation error at LINE 2 because method getFuelCapacity() is not overridden in class MaruthiCar

6406531562141. ✖ 

This program generates the output:

```
No petrol needed
Has 4 Wheels
```

6406531562142. ✓



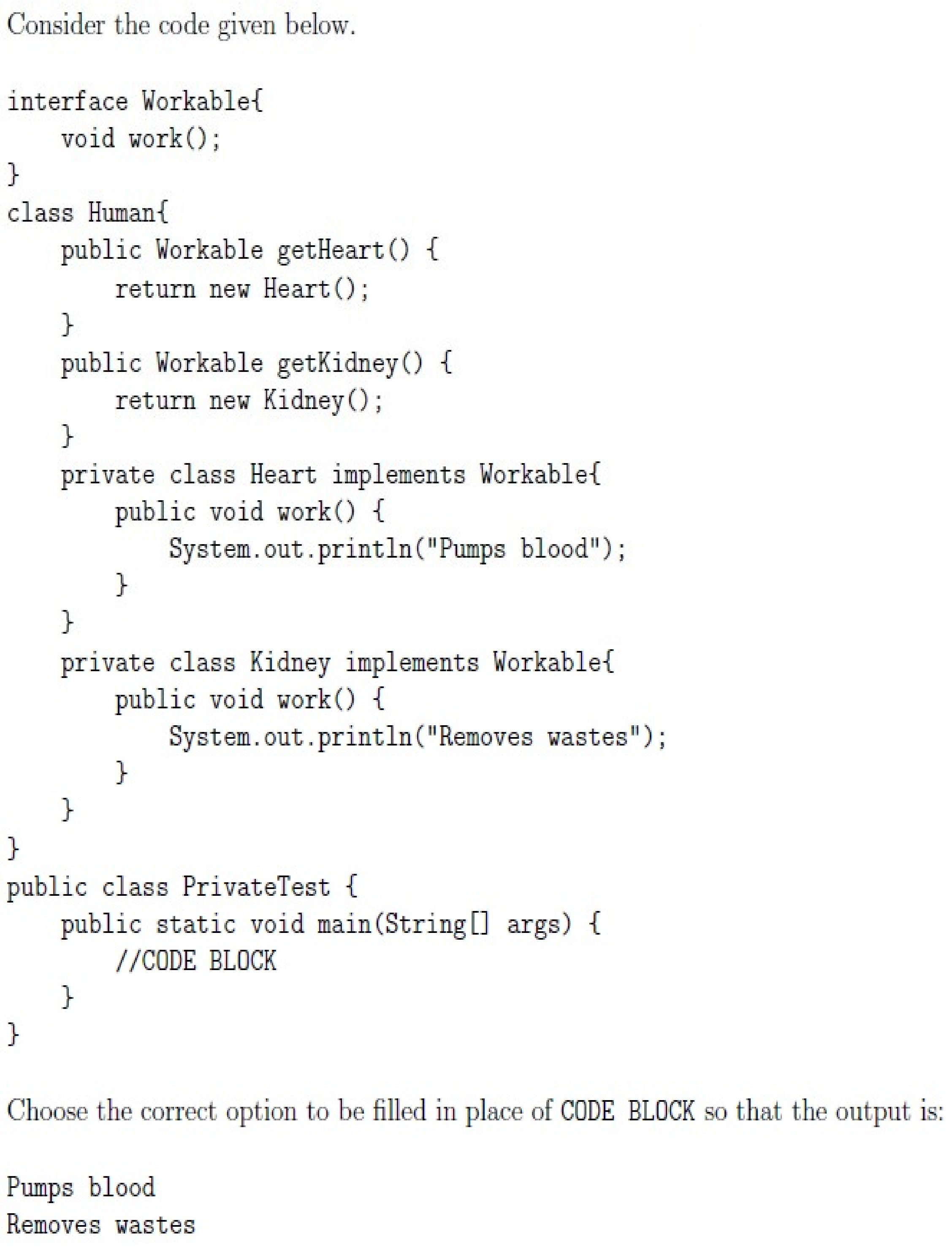
This program generates the output:

```
Has 2 Wheels
No petrol needed
Has 4 Wheels
Capacity: 20L Petrol
```

Question Number : 147 Question Id : 640653470076 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.

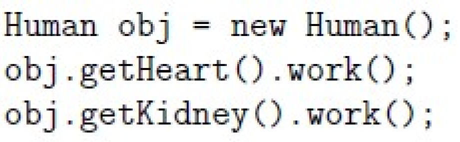
```
interface Workable{
    void work();
}
class Human{
    public Workable getHeart() {
        return new Heart();
    }
    public Workable getKidney() {
        return new Kidney();
    }
    private class Heart implements Workable{
        public void work() {
            System.out.println("Pumps blood");
        }
    }
    private class Kidney implements Workable{
        public void work() {
            System.out.println("Removes wastes");
        }
    }
}
public class PrivateTest {
    public static void main(String[] args) {
        //CODE BLOCK
    }
}
```

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

```
Pumps blood
Removes wastes
```

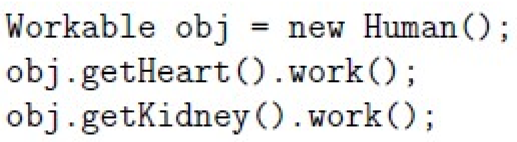
Options :

6406531562143. ✓



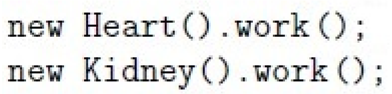
```
Human obj = new Human();
obj.getHeart().work();
obj.getKidney().work();
```

6406531562144. ✗



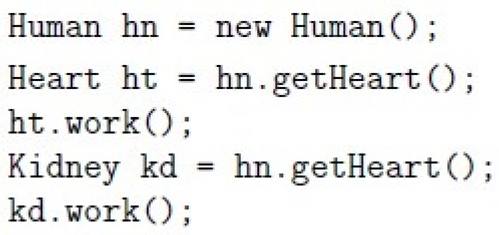
```
Workable obj = new Human();
obj.getHeart().work();
obj.getKidney().work();
```

6406531562145. ✗



```
new Heart().work();
new Kidney().work();
```

6406531562146. ✖



```
Human hn = new Human();
Heart ht = hn.getHeart();
ht.work();
Kidney kd = hn.getHeart();
kd.work();
```

Sub-Section Number : 4

Sub-Section Id : 64065367695

Question Shuffling Allowed : Yes

Is Section Default? : null

Question Number : 148 Question Id : 640653470063 Question Type : MSQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 3 Selectable Option : 0

Question Label : Multiple Select Question

Which of the following statements is/are correct?

Options :

6406531562091. ✖ Return value link points to the start of the previous activation record.

6406531562092. ✔ An activation record gets pushed into the stack when a function is called, and popped out when the function returns.

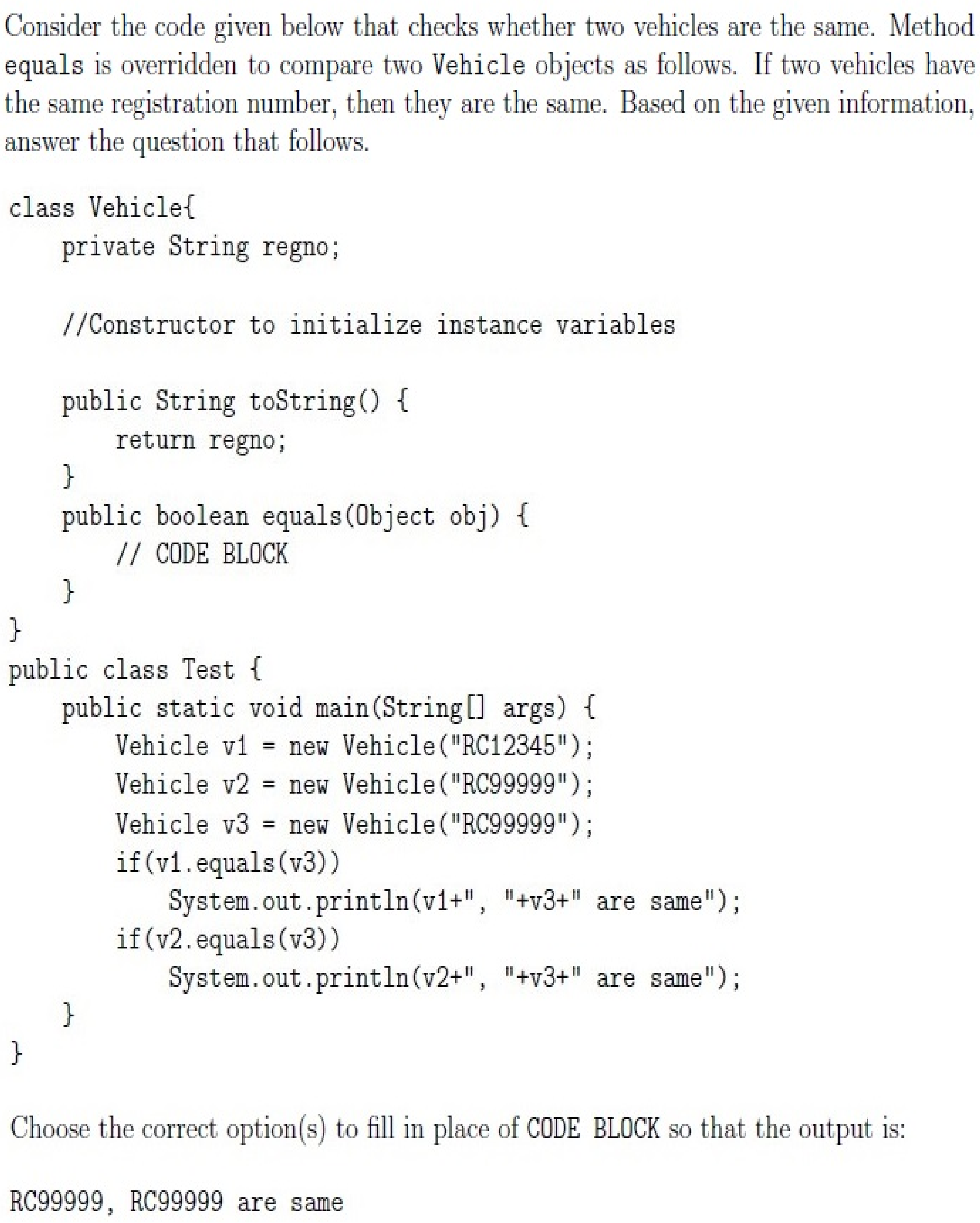
6406531562093. ✖ The variables present in every activation record in the stack are in scope and are accessible.

6406531562094. ✔ The variables present in the topmost activation record of the stack are in scope and are accessible.

Question Number : 149 Question Id : 640653470065 Question Type : MSQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 3 Selectable Option : 0

Question Label : Multiple Select Question



Consider the code given below that checks whether two vehicles are the same. Method equals is overridden to compare two Vehicle objects as follows. If two vehicles have the same registration number, then they are the same. Based on the given information, answer the question that follows.

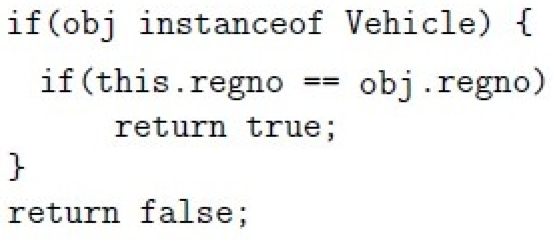
```
class Vehicle{
    private String regno;

    //Constructor to initialize instance variables

    public String toString() {
        return regno;
    }
    public boolean equals(Object obj) {
        // CODE BLOCK
    }
}
public class Test {
    public static void main(String[] args) {
        Vehicle v1 = new Vehicle("RC12345");
        Vehicle v2 = new Vehicle("RC99999");
        Vehicle v3 = new Vehicle("RC99999");
        if(v1.equals(v3))
            System.out.println(v1+", "+v3+" are same");
        if(v2.equals(v3))
            System.out.println(v2+", "+v3+" are same");
    }
}
```

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

```
RC99999, RC99999 are same
```

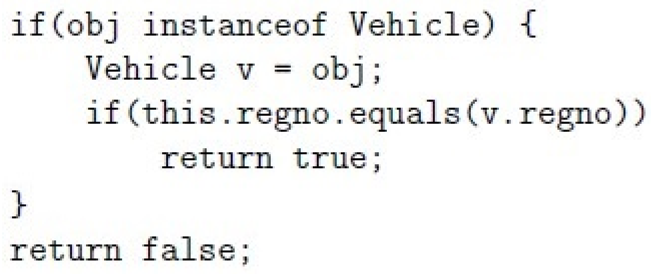
Options :

6406531562099. ✖



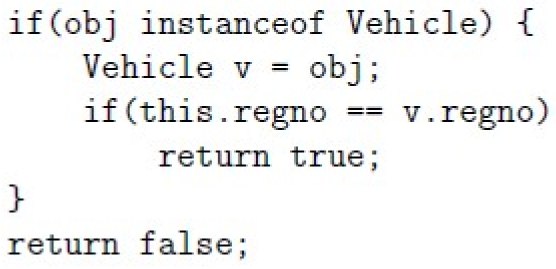
```
if(obj instanceof Vehicle) {
  if(this.regno == obj.regno)
      return true;
}
return false;
```

6406531562100. ✖



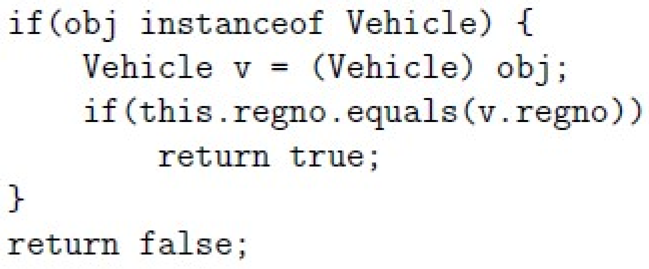
```
if(obj instanceof Vehicle) {
    Vehicle v = obj;
    if(this.regno.equals(v.regno))
        return true;
}
return false;
```

6406531562101. ✗



```
if(obj instanceof Vehicle) {
    Vehicle v = obj;
    if(this.regno == v.regno)
        return true;
}
return false;
```

6406531562102. ✓



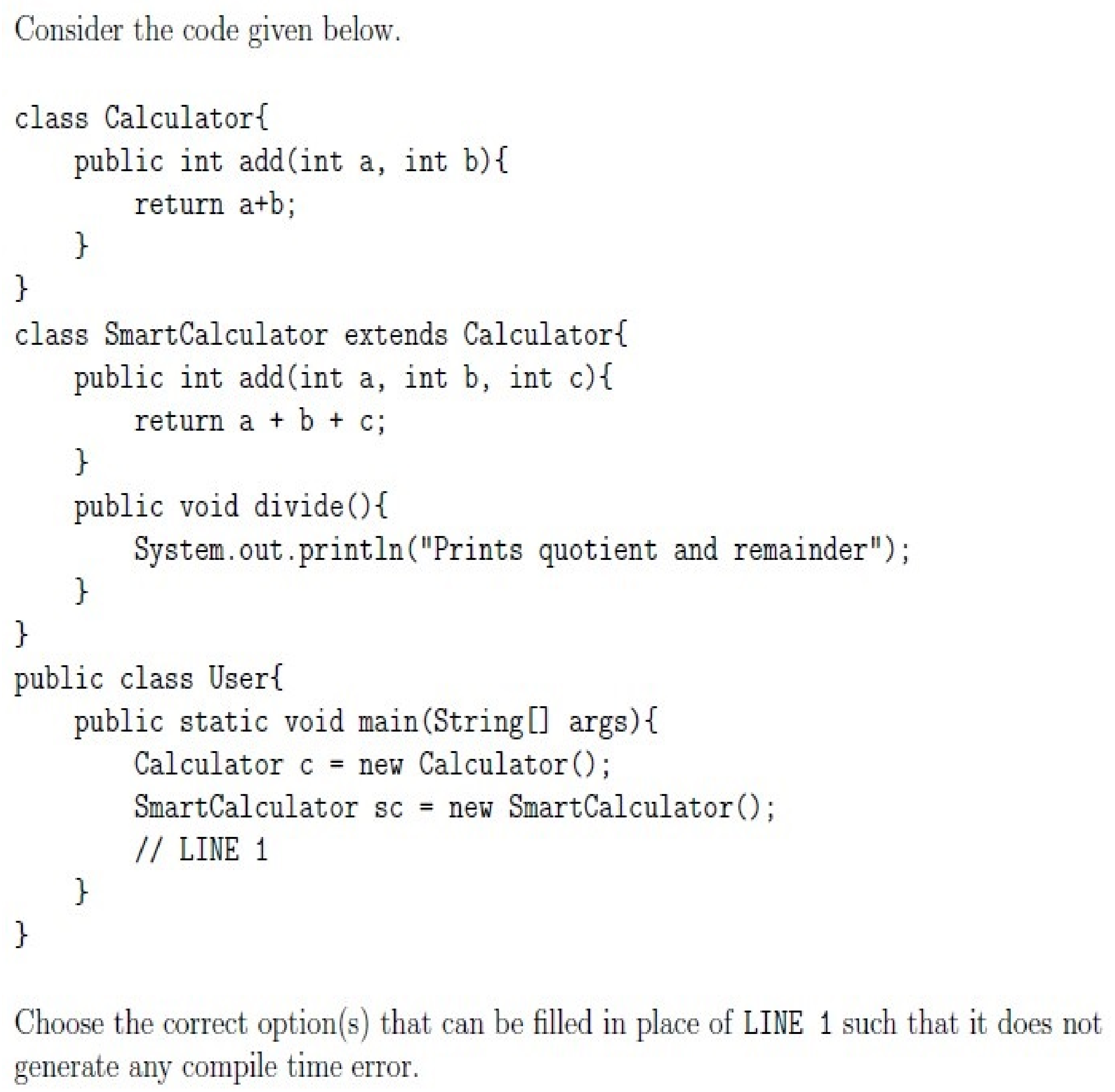
```
if(obj instanceof Vehicle) {
    Vehicle v = (Vehicle) obj;
    if(this.regno.equals(v.regno))
        return true;
}
return false;
```

Sub-Section Number :	5
Sub-Section Id :	64065367696
Question Shuffling Allowed :	Yes
Is Section Default? :	null

Question Number : 150 Question Id : 640653470067 Question Type : MSQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 4 Selectable Option : 0

Question Label : Multiple Select Question

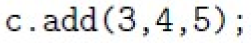


Consider the code given below.

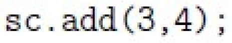
```
class Calculator{
    public int add(int a, int b){
        return a+b;
    }
}
class SmartCalculator extends Calculator{
    public int add(int a, int b, int c){
        return a + b + c;
    }
    public void divide(){
        System.out.println("Prints quotient and remainder");
    }
}
public class User{
    public static void main(String[] args){
        Calculator c = new Calculator();
        SmartCalculator sc = new SmartCalculator();
        // LINE 1
    }
}
```

Choose the correct option(s) that can be filled in place of LINE 1 such that it does not generate any compile time error.

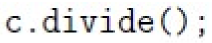
Options :

6406531562107. ✗ 

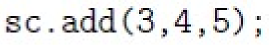
`c.add(3,4,5);`

6406531562108. ✓ 

`sc.add(3,4);`

6406531562109. ✗ 

`c.divide();`

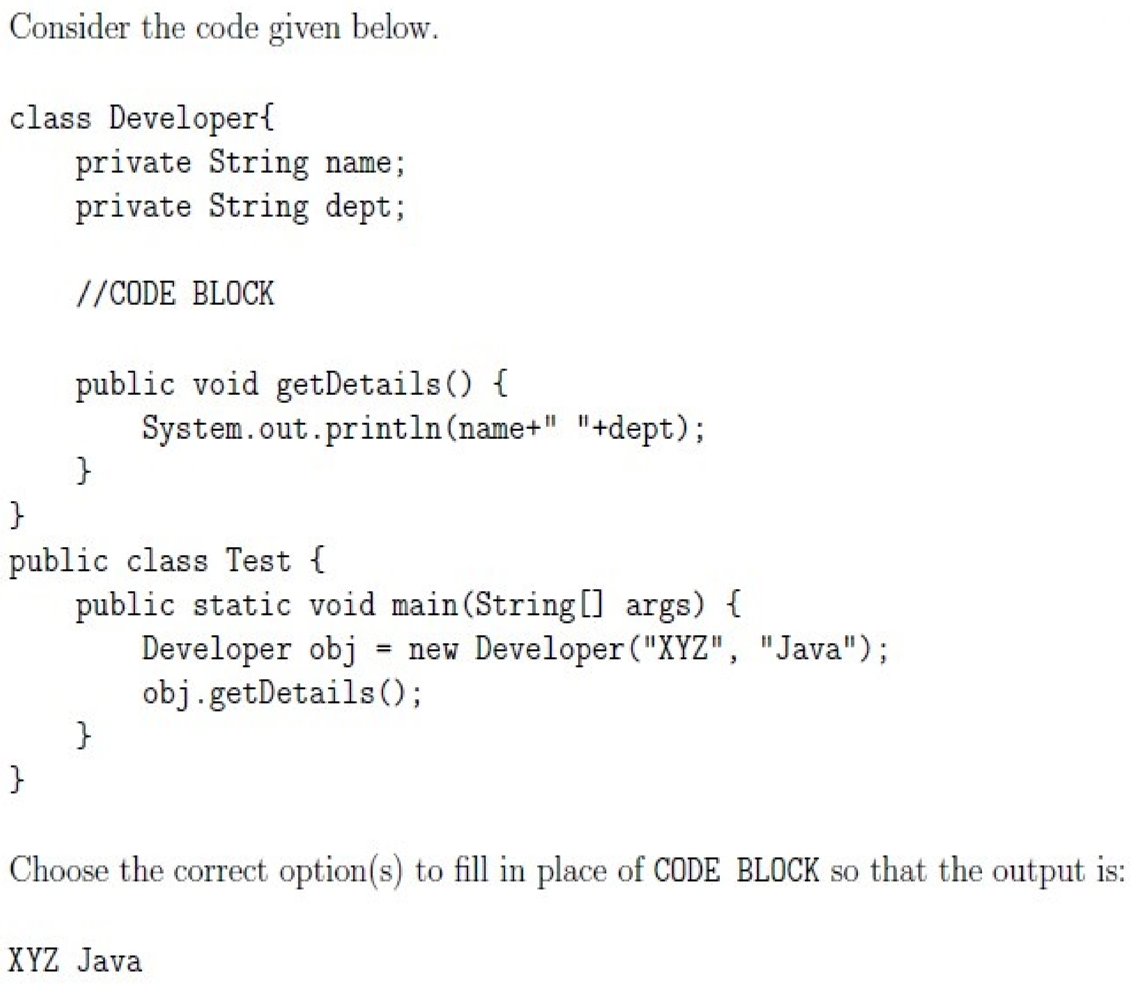
6406531562110. ✓ 

`sc.add(3,4,5);`

Question Number : 151 Question Id : 640653470070 Question Type : MSQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 4 Selectable Option : 0

Question Label : Multiple Select Question



Consider the code given below.

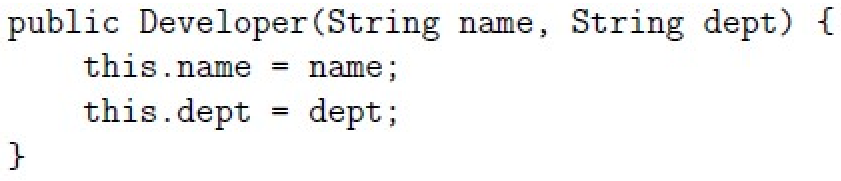
```
class Developer{
    private String name;
    private String dept;

    //CODE BLOCK

    public void getDetails() {
        System.out.println(name+" "+dept);
    }
}
public class Test {
    public static void main(String[] args) {
        Developer obj = new Developer("XYZ", "Java");
        obj.getDetails();
    }
}
```

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

```
XYZ Java
```

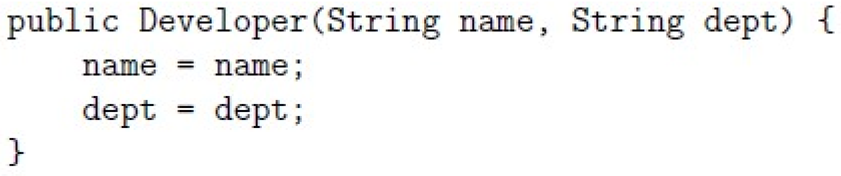
Options :

6406531562119. ✓



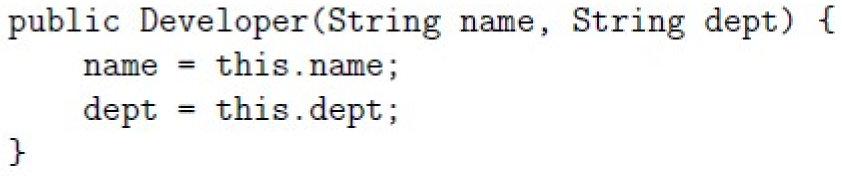
```
public Developer(String name, String dept) {
    this.name = name;
    this.dept = dept;
}
```

6406531562120. ✖



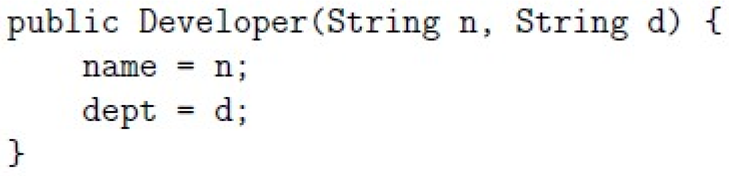
```
public Developer(String name, String dept) {
    name = name;
    dept = dept;
}
```

6406531562121. ✖



```
public Developer(String name, String dept) {
    name = this.name;
    dept = this.dept;
}
```

6406531562122. ✓



```
public Developer(String n, String d) {
    name = n;
    dept = d;
}
```

AppDev2

Section Id :	64065330336
Section Number :	10
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 :	64065367697
Question Shuffling Allowed :	No
Is Section Default? :	null

Question Number : 152 Question Id : 640653470078 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 0

Question Label : Multiple Choice Question

THIS IS QUESTION PAPER FOR THE SUBJECT "DIPLOMA LEVEL : MODERN APPLICATION DEVELOPMENT II"

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

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