

## Chapter 5 – Inheritance

### Question 1:

Given the code below,

```
1. package abcd;
2.
3. public class Dog
4. {
5.     protected static int count = 0;
6.     public Dog()
7.     {
8.         count++;
9.     }
10.    static int getRefCount()
11.    {
12.        return count;
13.    }
14.}
```

```
1. package singers;
2.
3. class Puppy extends abcd.Dog
```

```
4. {  
5.   Puppy()  
6.   {  
7.       count++;  
8.   }  
9.   public static void main(String args[])  
10.  {  
11.      System.out.print("BEFORE: " + count);  
12.      Puppy florence = new Puppy();  
13.      System.out.println(" AFTER: " + count);  
14.  }  
15. }
```

The output will be:

- A) Before:0 After:2
- B) Before:0 After:1
- C) Before:0 After:0
- D) Compilation will fail
- E) The compilation will succeed but an exception will be thrown.

Question 2:

Given the code below, What is the minimal modification that will make it compile correctly?

```
1.final class FirstClass
2.{
3.    int x;
4.    void y()
5.    {
6.        x = 1;
7.    }
8.}
9.
10.class SecondClass extends FirstClass
11.{
12.    final FirstClass FC = new FirstClass();
13.    final void y()
13.    {
14.        System.out.println("In method y()");
15.        FC.x = 12345;
16.    }
17.}
```

- ( ) A) remove the final modifier from FirstClass declaration
- ( ) B) remove the final modifier from FirstClass declaration as well as from the variable FC in

SecondClass

- ( ) C) No modification is needed
- ( ) D) remove the final modifier from the variable FC in SecondClass

Question 3:

Which of the following statements is true ?

- A) It is possible to have a final class with an abstract method
- B) An abstract method can't be static

Question 4:

Given the code below, and making no other changes,

```
1. class SuperDuper
2. {
3.     void zzz() { }
4. }
5.
6. class Sub extends SuperDuper
7. {
8.     void zzz() { }
```

9. }

Which of the following statements is true ?

- A) It is legally possible to place 'public' before zzz on line 3
- B) It is legally possible to place 'private' before zzz on line 3

Question 5:

Consider the following classes,

```
1. public class Base
2. {
3.     public void method(int j)
4.     {
5.         System.out.println("Tel-Aviv");
6.     }
7. }
```

```
1. public class Derive1 extends Base
2. {
3.     public void method(int j)
4.     {
5.         System.out.println("Haifa");
```

6. }

7. }

1. public class Tester

2. {

3.     public static void main(String args[])

4.     {

5.         Base b1 = new Base();

6.         Base b2 = new Derive1();

7.         b1.method(5);

8.         b2.method(6);

9.     }

10.    public void method(int j)

11.    {

12.        System.out.println("Jerusalem");

13.    }

14. }

What output results when the main method of the class Tester is run?

A) Tel-Aviv

Haifa

B) Tel-Aviv

Tel-Aviv

C) Haifa

- Haifa
- D) Haifa
- Tel-Aviv
- E) Jerusalem
- Jerusalem

Question 6:

Consider the following classes definition,

```
1. public class Derive extends Base
2. {
3.     public Derive(String str)
4.     {
5.     }
6.     public Derive(int j, int k)
7.     {
8.         super(j, k);
9.     }
10.}
```

Which of the following forms of constructor must exist explicitly in the definition of the Base class?

- A) Base()
- B) Base(String str)
- C) Base(int num1, int num2)
- D) Base(char number1, char number2)
- E) Base(int num1, int num2, int num3)

Question 7:

You have been given a design document for cars registration system .

The design document states:

"A car has an owner, a manufacturing date and a registration number. A private car is a car that has a flag indicating if it has been in an Accident , and a textual description of its flows."

Given that the Car class has already been defined, which of the following fields would be appropriate for inclusion in the PrivateCar class as members?

- A) boolean accident
- B) String strFlowDescription
- C) int registrationNumber
- D) Date manufacturingDate
- E) String ownerName



Question 8:

Consider these classes, defined in separate source files:

```
1. public class Base
2. {
3.     public float method(float a) throws IOException {}
4. }
```

```
1. public class Derive extends Base
2. {
3.
4. }
```

Which of the following methods would be legal (individually) at line 3 in class Derive ?

- A) public float method (float a) {}
- B) float method (float a) {}
- C) public float method (float a) throws SQLException {}
- D) public float method (int num) throws SQLException {}
- E) public float method (float a, float b) throws SQLException {}

Question 9:

Which of the following declaration are legal ?

- A) abstract final double getSize();
- B) abstract int size;
- C) abstract friendly String getName()
- D) public float method (int num) throws SQLException {}
- E) public float method (float a, float b) throws SQLException {}

Question 10:

Which of the following statements is true ?

- A) It is possible to have a final class with an abstract method
- B) An abstract method can't be final

Question 11:

Consider these classes, defined in separate source files:

1. final class Base

2.{

```

3.    final Point p = new Point();
4.                protected float method(float a) throws IOException
5.                {
6.                    if (a<0)
7.                        throw new IOException();
8.                    return 1.0f;
9.                }
5.}

```

```

1. class Derive extends Base
2. {
3.    float method(float b) throws IOExcpeion
4.    {
5.                p.x = 12;
6.                if (b<0)
7.                    throw new IOException();
8.    return p.x * p.y;
9.    }
10.}

```

```

1. class Point
2. {
3.    float x,y;
4. }

```

What are the minimal changes that will make the code compile correctly?

- A) On line 1 in class Base remove the final modifier
- B) On line 3 in class Base remove the final modifier
- C) On line 4 in class Base remove the protected modifier
- D) On line 3 in class Derive change the parameter name from b to a
- E) On line 3 in class Point change the variables type from float to int

Question 12:

Given the code below, which access modifiers (public, protected or private) can legally be placed before shemeshMethod on line 3 in class Base declaration ?

```
1. class Base
2. {
3.     int shemeshMethod() {return 2000;}
4. }
```

```
1. class Derive extends Base
2. {
3.     protected int shemeshMethod() {return 2001;}
4. }
```

- A) private
- B) public
- C) protected

Question 13:

Which of the following statements is true ?

- A) The rules that govern converting object references are identical in both method calls and assignments
- B) The rules that govern converting object references are not identical in method calls and in assignments.

Question 14:

Given the following class definition:

```
1. public class Derived extends Base
2. {
3.     public Derived() {}
4.     public Derived(int num)
5.     {
6.         this.num = num;
```

```

7.   }
8.   public Derived(int a, int b)
9.   {
10.      this.num = a+b;
11.  }

```

What are the constructors that must be explicitly declared in Base class ?

- A) Base()
- B) Base(int j)
- C) Base(int m, int n)
- D) Base(String num)\_
- E) Base(char [] vec)

Question 15:

Consider these classes, defined in separate source files:

```

1. public class Base
2. {
3.     public float method(float a) throws IOException {}
4. }

```

```

1. public class Derive extends Base

```

2. {

3.

4. }

Which of the following methods would be legal (individually) at line 3 in class Derive ?

- A) public float method (float a) {} throws Exception
- B) float method (float a) {}
- C) protected double method (float a, int b) throws IOException {}
- D) public float method (int num) throws SQLException {}
- E) public float method (float a, float b) throws SQLException {}

Question 16:

True or False:

Methods can't be overridden and get an access modifier, which is more private.

- A) True
- B) False

Question 17:

True or False:

If the derived class constructor has a call to a specific base class constructor using the super keyword then the call must be placed at the first line in the derived class constructor.

- A) True
- B) False

Question 18:

Given the code below, and making no other changes,

1. class SuperDuper
2. {
3. void zzz() { }
4. }
- 5.
6. class Sub extends SuperDuper
7. {
8. void zzz() { }
9. }



It is not legally possible to place 'protected' before zzz on line 3.

- A) True
- B) False

Question 19:

Given the following class definition:

```
1. public class Derived extends Base
2. {
3.     public Derived(int num)
4.     {
5.         super(num);
6.         this.num = num;
7.     }
8.     public Derived(int a, int b)
9.     {
10.        this.num = a+b;
11.    }
```

What are the constructors that must be explicitly declared in Base class ?

- A) Base()
- B) Base(int j)
- C) Base(int m, int n)
- D) Base(String num)\_
- E) Base(char [] vec)

Question 20:

Given the code below, and making no other changes,

```
1. class SuperDuper
2. {
3.     void zzz() { }
4. }
5.
6. class Sub extends SuperDuper
7. {
8.     private void zzz() { }
9. }
```

It is legally possible to place 'public' before zzz on line 3

- A) True
- B) False

Question 21:

True or False:

Methods can't be overridden and get an access modifier which is more public.

- A) True
- B) False

Question 22:

Given the code below:

```
1. class Try1
2. {
3.     int luckyNum;
4.     public void tutu()
5.     {
6.         luckyNum = 9;
7.     }
8. }

9. class Try1D extends Try1
10. {
```

```
11.     void tutu(int num)
12.     {
13.         luckyNum = num;
14.     }
15. }

16. public class TryApp
17. {
18.     public static void main(String args[])
19.     {
20.         Try1D tObj = new Try1D();
21.         tObj.tutu(10);
22.         System.out.println("luckyNum = " + tObj.luckyNum);
23.     }
24. }
```

The output is:

- A) 10
- B) 9
- C) The code doesn't compile
- D) The output depends in the platform on which the program runs

Question 23:

True or False:

Java allows a class to implements only one interface.

- A) True
- B) False

Question 24:

True or False:

An interface can extend one (or more) other interface's.

- A) True
- B) False

Question 25:

Given the code below:

```
1.  interface Computer
2.  {
3.      public int computeSum(int num1, int num2);
4.  }
5.  class TV implements Computer
6.  {
7.      public int computeSum(int num1, int num2)
8.      {
9.          return num1+num2;
10.     }
11.     public void openTV()
12.     {
13.         System.out.println("The TV is open!");
14.     }
15. }
16. public class TryApp
17. {
18.     public static void main(String args[])
19.     {
20.         Computer myComputer = new TV();
21.         myComputer.openTV();
```

22.        }

23.    }

The output is:

- A) The TV is open.
- B) The code doesn't compile.
- C) The output depends in the platform on which the program runs.
- D) The TV is off.

Question 26:

True or False:

A class can't extend more than one class.

- A) True
- B) False

Question 27:

True or False:

An interface and an abstract class are the same.

- A) True
- B) False

Question 28:

Given the code below:

```
1.class Try100
2.{
3.    static int nony = 7;
4.    public int getLuck()
5.    {
6.        return nony;
7.    }
8.    public static void main(String args[])
9.    {
10.        Try100 t = new Try100D();
11.        System.out.println("The Lucky number is : " + t.getLuck());
```



12. }

13.}

14. class Try100D extends Try100

15. {

16.     static int nony = 8;

17.     public int getLuck()

18.     {

19.         return nony;

20.     }

21.}

- ( ) A) The output will be : "The Lucky number is : 8"
- ( ) B) The output will be : "The Lucky number is : 9"
- ( ) C) The code won't compile
- ( ) D) The code will compile but won't execute properly

Question 29:

Given the code below:

1.     public class LittleExtend

2.     {

3.         public static void main(String args[])

```
4.      {  
5.          Child child = new Child();  
6.          System.out.println(child.getNum());  
7.      }  
8.  }
```

```
9.  class Father  
10. {  
11.     private final int getNum(int i)  
12.     {  
13.         return 10;  
14.     }  
15. }
```

```
16. class Child extends Father  
17. {  
18.     int getNum()  
19.     {  
20.         return 20;  
21.     }  
22. }
```

The output is:

- A) 20
- B) 10
- C) 0
- D) The code doesn't compile successfully

Question 30:

Given the code below,

```
1. package abcd;
2.
3. public class Dog
4. {
5.     protected int count = 0;
6.     public Dog()
7.     {
8.         count++;
9.     }
10.    static int getRefCount()
11.    {
12.        return count;
13.    }
14.}
```

```
1. package singers;
2.
3. class Puppy extends abcd.Dog
4. {
5.     Puppy()
6.     {
7.         count++;
8.     }
9.     public static void main(String args[])
10.    {
11.        System.out.print("BEFORE: " + count);
12.        Puppy florence = new Puppy();
13.        System.out.println(" AFTER: " + count);
14.    }
15. }
```

The output will be:

- A) Before:0 After:2
- B) Before:0 After:1
- C) Before:0 After:0
- D) Compilation will fail
- E) The compilation will succeed but an exception will be thrown.

Question 31:

Given the following class definition:

```
1. public class Derived extends Base
2. {
3.     public Derived(int num)
4.     {
5.         super(num);
6.         this.num = num;
7.     }
8.     public Derived(int a, int b)
9.     {
10.        super(a, b);
11.        this.num = a+b;
12.    }
```

What are the constructors that must be explicitly declared in Base class ?

- A) Base()
- B) Base(int j)
- C) Base(int m, int n)
- D) Base(String num)\_
- E) Base(char [] vec)

Question 32:

Given the code below, which access modifiers (public, protected or private) can legally be placed before shemeshMethod on line 3 in class Derive declaration ?

1. class Base

2. {

3.     int shemeshMethod() {return 2000;}

4. }

1. class Derive extends Base

2. {

3.     int shemeshMethod() {return 2001;}

4. }

- A) private
- B) public
- C) protected

Question 33:

Given the code below, and making no other changes,

```
1. class SuperDuper
2. {
3.     void zzz() { }
4. }
5.
6. class Sub extends SuperDuper
7. {
8.     void zzz() { }
9. }
```

Which of the following statements is true ?

- A) It is legally possible to place 'public' before zzz on line 8
- B) It is legally possible to place 'private' before zzz on line 8
- C) It is legally possible to place 'protected' before zzz on line 8

Question 34:

Given the code below:

```
23. public class LittleExtend
24. {
25.     public static void main(String args[])
26.     {
27.         Father child = new Child();
28.         System.out.println(child.getNum());
29.     }
30. }

31. class Father
32. {
33.     int getNum()
34.     {
35.         return 10;
36.     }
37. }

38. class Child extends Father
39. {
40.     int getNum()
41.     {
```



```
42.         return 20;
43.     }
44. }
```

The output is:

- A) 20
- B) 10
- C) 0
- D) The code doesn't compile successfully

Question 35:

Given the following class definition:

```
1. public class Derived extends Base
2. {
3.     public Derived(int num)
4.     {
5.         this.num = num;
6.     }
7.     public Derived(int a, int b)
8.     {
9.         this.num = a+b;
```

10. }

What are the constructors that must be explicitly declared in Base class ?

- A) Base(String str)
- B) Base(int j)
- C) Base(int m, int n)
- D) Base(String num)\_
- E) Base(char [] vec)

Question 36:

Given the code below, which access modifiers (public, protected or private) can legally be placed before shemeshMethod on line 3 in the class Derive declaration ?

```
1. class Base
2. {
3.     int shemeshMethod() {return 2000;}
4. }
```

```
1. class Derive extends Base
2. {
3.     int shemeshMethod(int num) {return 2001*num;}
4. }
```

- A) private
- B) public
- C) protected

Question 37:

Given the code below, and making no other changes,

```
1. class SuperDuper
2. {
3.     private void zzz() { }
4. }
5.
6. class Sub extends SuperDuper
7. {
8.     void zzz() { }
9. }
```

Which of the following statements is true ?

- A) It is legally possible to place 'public' before zzz on line 8
- B) It is legally possible to place 'private' before zzz on line 8
- C) It is legally possible to place 'protected' before zzz on line 8

Question 38:

Given the code below:

```
45. public class LittleExtend
46. {
47.     public static void main(String args[])
48.     {
49.         Father child = new Child();
50.         System.out.println(child.getNum());
51.     }
52. }
```

```
53. class Father
54. {
55.     protected int getNum()
56.     {
57.         return 10;
58.     }
59. }
```

```
60. class Child extends Father
61. {
62.     int getNum()
63.     {
```

```
64.         return 20;
65.     }
66. }
```

The output is:

- A) 20
- B) 10
- C) 0
- D) The code doesn't compile successfully

Question 39:

Given the following class definition:

```
1. public class Derived extends Base
2. {
3.     public Derived(int num)
4.     {
5.         this.num = num;
6.     }
7.     public Derived(int a, int b)
8.     {
9.         super(a,b);
```

```
10.     this.num = a+b;
11. }
```

What are the constructors that must be explicitly declared in Base class ?

- A) Base(String str)
- B) Base(int j)
- C) Base(int m, int n)
- D) Base(String num)\_
- E) Base(char [] vec)
- F) Base()

Question 40:

Given the following class definition:

```
1. public class Something extends Base
2. {
3.     public Something(int i)
4.     {
5.     }
6.     public Something(int i, int j)
7.     {
8.         super(i, j);
9.     }
10.    public Something(int i, int j, int k)
```

```

11. {
12.     super(i, j, k);
13. }
10.}

```

Which of the following forms of constructors must exist explicitly in the definition of the Base class ?

- A) Base()
- B) Base(byte b1, byte b2)
- C) Base(int num)
- D) Base(int num, short number)
- E) Base(short num1, short num2, short num3)

Question 41:

Given the following class definition:

```

1. public class Something extends Base
2. {
3.     public Something(int i)
4.     {
5.     }
6.     public Something(int i, int j)
7.     {
8.         super(i, j);
9.     }
10.    public Something(int i, int j, int k)
11.    {
12.        super(i, j, k);

```

```
13. }  
10.}
```

Which of the following forms of constructors must exist explicitly in the definition of the Base class ?

- A) Base()
- B) Base(byte b1, byte b2)
- C) Base(int num)
- D) Base(int num, short number)
- E) Base(int num1, int num2, int num3)

Question 42:

Given the following class definition:

```
1. public class Something  
2. {  
3.     public static final int BLANK=0, X=1, O=2;  
4.     public static void main(String args[])  
5.     {  
6.         X=8;  
7.         System.out.println("X="+X);  
8.     }  
9. }
```

The output is:

- ( ) A) X=8



- ( ) B) X=1
- ( ) C) The code doesn't compile
- ( ) D) An exception is thrown in runtime.

Question 43:

Given the code below,

```
1. package animals;
2.
3. class Dog
4. {
5.     protected static int count = 0;
6.     public Dog()
7.     {
8.         count++;
9.     }
10.    static int getRefCount()
11.    {
12.        return count;
13.    }
14.}
```

  

```
1. package animals.children;
```

```
2.  
3. public class Puppy  
4. {  
5.     static int count = 2;  
6.     Puppy()  
7.     {  
8.         count++;  
9.     }  
10.    public static void main(String args[])  
11.    {  
12.        System.out.print("BEFORE: " + count);  
13.        Puppy florence = new Puppy();  
14.        System.out.println(" AFTER: " + count);  
15.    }  
16. }
```

The output will be:

- A) Before:2 After:2
- B) Before:2 After:3
- C) Before:2 After:0
- D) Compilation will fail
- E) The compilation will succeed but an exception will be thrown.

Question 44:

Given the code below, What is the minimal modification that will make it compile correctly?

```
1.class FirstClass
2.{
3.    int x;
4.    final void y()
5.    {
6.        x = 1;
7.    }
8.}
9.
10.class SecondClass extends FirstClass
11.{
12.    final FirstClass FC = new FirstClass();
13.    void y()
13.    {
14.        System.out.println("In method y()");
15.        FC.x = 12345;
16.    }
17.}
```

- ( ) A) remove the final modifier from the variable x in the FirstClass declaration
- ( ) B) remove the final modifier from the variable x in the FirstClass declaration as well as from the variable FC in SecondClass
- ( ) C) No modification is needed
- ( ) D) remove the final modifier from the variable FC in SecondClass
- ( ) E) remove the final modifier from the method y() in the FirstClass declaration.

Question 45:

Which of the following statements is true ?

- A) It is not possible to have a final class with an abstract method
- B) It is possible to have an abstract method, which is static

Question 46:

Given the code below, and making no other changes,

1. class SuperDuper
2. {
3. void zzz() { }
4. }
- 5.

```
6. class Sub extends SuperDuper
7. {
8.     void zzz() { }
9. }
```

Which of the following statements is true ?

- A) It is legally possible to place 'public' before zzz on line 8
- B) It is legally possible to place 'private' before zzz on line 8
- C) It is legally possible to place 'protected' before zzz on line 8
- D) It is legally possible to place 'private' before zzz on line 3
- E) It is legally possible to place 'protected' before zzz on line 3
- F) It is legally possible to place 'public' before zzz on line 3

Question 47:

Consider the following classes,

```
1. public class Base
2. {
3.     public int num = 100;
4.     public void method(int j)
5.     {
6.         num+=j;
```

7. }

8. }

1. public class Derive1 extends Base

2. {

3.     public void method(int j)

4.     {

5.         num\*=j;

6.     }

7. }

1. public class Tester

2. {

3.     public int num;

3.     public static void main(String args[])

4.     {

5.         Base b1 = new Base();

6.         Base b2 = new Derive1();

7.         b1.method(5);

8.         b2.method(6);

9.         System.out.println(b1.num);

10.        System.out.println(b2.num);

11.     }

12.     public void method(int j)

13.     {

14.        num-=j;

15.    }

16.}

What output results when the main method of the class Tester is run?

A) 105  
600

B) 500  
106

C) 95  
94

D) 500  
600

E) 105  
106

Question 48:

Consider the following classe definition,

```
1. public class Derive extends Base
2. {
3.     public Derive(String str)
4.     {
5.     }
6.     public Derive(int j, int k)
7.     {
8.         super(k);
9.     }
10.}
```

Which of the following forms of constructor must exist explicitly in the definition of the Base class?

- A) Base()
- B) Base(String str)
- C) Base(int num1, int num2)
- D) Base(char number1, char number2)
- E) Base(int num1, int num2, int mum3)
- F) Base(int number)



Question 49:

You have been given a design document for cars registration system.

The design document states:

"A car has an owner, a manufacturing date, a color and a registration number. A private car is a car that has a flag indicating if it has been in an accident."

Given that the Car class has already been defined, which of the following fields would be appropriate for inclusion in the PrivateCar class as members?

- A) boolean accident
- B) String strFlowDescription
- C) int registrationNumber
- D) Date manufacturingDate
- E) String ownerName
- F) Color color

Question 50:

Consider these classes, defined in separate source files:

```
1. public class Base
2. {
3.     public float method(float a) {}
4. }
```

```
1. public class Derive extends Base
2. {
3.
4. }
```

Which of the following methods would be legal (individually) at line 3 in class Derive ?

- A) public float method (float a) {}
- B) float method (float a) {}
- C) private float method (float a) {}
- D) float method (int num) {}
- E) public float method (float a, float b) {}

Question 51:

Which of the following declaration are legal ?

- A) abstract final double getSize();
- B) abstract int size;
- C) abstract friendly String getName()
- D) public float method (int num) {}
- E) public float method (float a, float b) {}

Question 52:

Which of the following statements is true ?

- A) It is possible to have a final class with an abstract variable
- B) An abstract method can't be final

Question 53:

Which of the following statements is true ?

- A) Static methods and final methods cannot be overridden.
- B) Static methods can be overridden but final methods can't be.

Question 54:

What is the output?

```
1. class Calculator
2. {
3.     static int getNum()
4.     {
5.         return 7;
6.     }
7.     int getValue()
8.     {
9.         return 5;
10.    }
11. }

12. class GreatCalculator extends Calculator
13. {
14.     static int getNum()
15.     {
16.         return 12;
17.     }
18.     int getValue()
19.     {
20.         return 4;
```

21. }

22. }

23. public class CalculusDemo

24. {

25.     public static void main(String args[])

26.     {

27.         Calculator calc = new GreatCalculator();

28.         System.out.println((calc.getNum()+calc.getValue()));

29.     }

30. }

- A) 11
- B) 19
- C) 17
- D) 9
- E) The code doesn't compile.

Question 55:

Given the following code:

```
1.  public class Demo
2.  {
3.      public static void main(String args[])
4.      {
5.          Car car = new SportCar();
6.          System.out.println(car.brand + "," + car.volume);
7.      }
8.  }
9.
10.
11. class Car
12. {
13.     int volume;
14.     String brand = "TOYOTA";
15.     Car()
16.     {
17.         this(1600);
18.     }
19.     Car(int vol)
20.     {
21.         volume = vol;
```

```
22.     }  
23. }  
24.  
25.  
26. class SportCar extends Car  
27. {  
28.     int volume;  
29.     String brand = "LAMBORGINI";  
30.     SportCar()  
31.     {  
32.         this(3000);  
33.     }  
34.     SportCar(int vol)  
35.     {  
36.         volume = vol;  
37.     }  
38. }
```

What is the output?

- ( ) A) TOYOTA,1600
- ( ) B) LAMBORGINI,3000
- ( ) C) FIAT,1600
- ( ) D) TOYOTA,3000
- ( ) E) LAMBORGINI,1600

Question 56:

Given the following code:

```
1.  public class Demo
2.  {
3.      public static void main(String args[])
4.      {
5.          Car car = new SportCar();
6.          System.out.println(((SportCar)car).brand + "," + ((SportCar)car).volume);
7.      }
8.  }
9.
10.
11. class Car
12. {
13.     int volume;
14.     static String brand = "TOYOTA";
15.     Car()
16.     {
17.         this(1600);
18.     }
19.     Car(int vol)
20.     {
21.         volume = vol;
```



```
22.     }
23. }
24.
25.
26. class SportCar extends Car
27. {
28.     int volume;
29.     static String brand = "LAMBORGINI";
30.     SportCar()
31.     {
32.         this(3000);
33.     }
34.     SportCar(int vol)
35.     {
36.         volume = vol;
37.     }
38. }
```

What is the output?

- ( ) A) TOYOTA,1600
- ( ) B) LAMBORGINI,3000
- ( ) C) FIAT,1600
- ( ) D) TOYOTA,3000
- ( ) E) LAMBORGINI,1600

Question 57:

Given the following code:

```
39. public class Demo
40. {
41.     public static void main(String args[])
42.     {
43.         Car car = new SportCar();
44.         System.out.println(car.getNum());
45.     }
46. }
47.
48.
49. class Car
50. {
51.     int volume;
52.     static String brand = "TOYOTA";
53.     Car()
54.     {
55.         this(1600);
56.     }
57.     Car(int vol)
58.     {
59.         volume = vol;
```

```
60.     }
61.     int getNum()
62.     {
63.         int numA =8;
64.         numA = 8 + numB;
65.         int numB = 2;
66.         return numA;
67.     }
68. }
69.
70.
71. class SportCar extends Car
72. {
73.     int volume;
74.     static String brand = "LAMBORGINI";
75.     SportCar()
76.     {
77.         this(3000);
78.     }
79.     SportCar(int vol)
80.     {
81.         volume = vol;
82.     }
83. }
```

The output is:

- A) 10
- B) 12
- C) The code won't compile
- D) 8
- E) 2