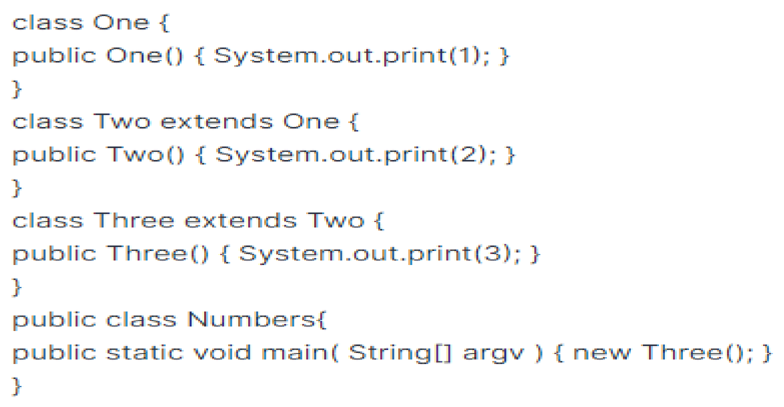
The following code snippet got error(s). Identify the line no(s) which got the error(s).
Note that the classes Pen, Paper and Pencil have been coded prior and available.

```
 1. public class Clerk {
 2. // code omitted
 3. public void checkFiles (Paper paper, Pen pen) {
 4. // code omitted
 5. }
 6. }
 7. // new class
 8. public class Manager {
 9. private Clerk ck;
10. private Pen _pen;
11. // code omitted
12. public void giveClerkWork (Paper paper) {
13. ck.checkFiles (pen, paper);
14. }
15. }
```

a) Line No 3
b) Line No 13
c) Line No 12
d) Line No 3 and 13
e) Line No 3 and 12

Q2 What is the result when this code is executed?



```
class One {
public One() { System.out.print(1); }
}
class Two extends One {
public Two() { System.out.print(2); }
}
class Three extends Two {
public Three() { System.out.print(3); }
}
public class Numbers{
public static void main( String[] argv ) { new Three(); }
}
```

a) 3
b) 321
c) 123
d) The code runs with no output
e) error

Q3 What will be the output of the following code, if the input is 9001 ?

```
import java.util.*;
public class Q1 {
    public static void main(String[] args) {
        {
 Scanner sc = new Scanner(System.in);
 System.out.print("Enter an integer: ");
 int n = sc.nextInt();
 System.out.println("Method APU: " + APU(n));
 System.out.println("Method APIIT: " + APIIT(n));
 System.out.println("Method APIS: " + Integer.parseInt(APIS(n)));
         }   }

public static int APU (int n) {
 int sum = 0;
 while (n > 0) {
 sum += n % 10;
 n /= 10;
               }
 return sum;
                              }
 public static int APIIT (int n) {
 int n1 = n;
 while (n1 >= 10) {
 n1 = 0;
 while (n > 0) {
 n1 += n % 10;
 n /= 10;
              }
 n = n1;
             }
 return n1;
                                 }
 public static String APIS (int n) {
 String s = new Integer(n).toString();
 String s1 = "";
 int len = s.length();
 for (int i = 0; i < len; i++)
 s1 += s.substring(len-i-1, len-i);
 return s1;
                                   }
                }
```

a)

```
Method APU: 10
Method APIIT: 1009
Method APIS: 1
```

b)

```
Method APU: 1
Method APIIT: 10
Method APIS: 1009
```

c)

```
Method APU: 10
Method APIIT: 1
Method APIS: 1009
```

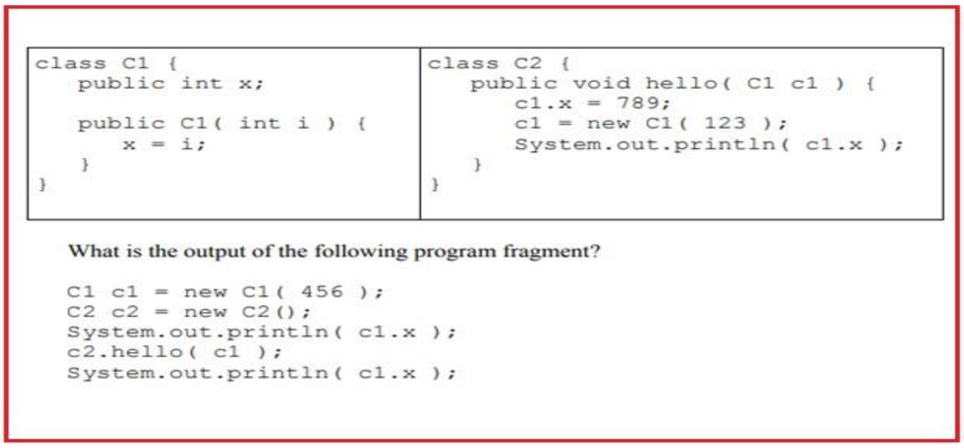
d)

```
Method APU: 1009
Method APIIT: 1
Method APIS: 10
```

e)

None

Q4 Given the following classes C1 and C2.



```
class C1 {
   public int x;

   public C1( int i ) {
      x = i;
   }
}
```

```
class C2 {
   public void hello( C1 c1 ) {
      c1.x = 789;
      c1 = new C1( 123 );
      System.out.println( c1.x );
   }
}
```

What is the output of the following program fragment?

```
C1 c1 = new C1( 456 );
C2 c2 = new C2();
System.out.println( c1.x );
c2.hello( c1 );
System.out.println( c1.x );
```

a)

```
456
123
789
```

b)

```
456
789
123
```

c)

```
123
789
456
```

d)

```
789
456
123
```

e)

```
456
123
456
```

Q5 What will be the output of the following code if 6,9,12,9,15 are the inputs?

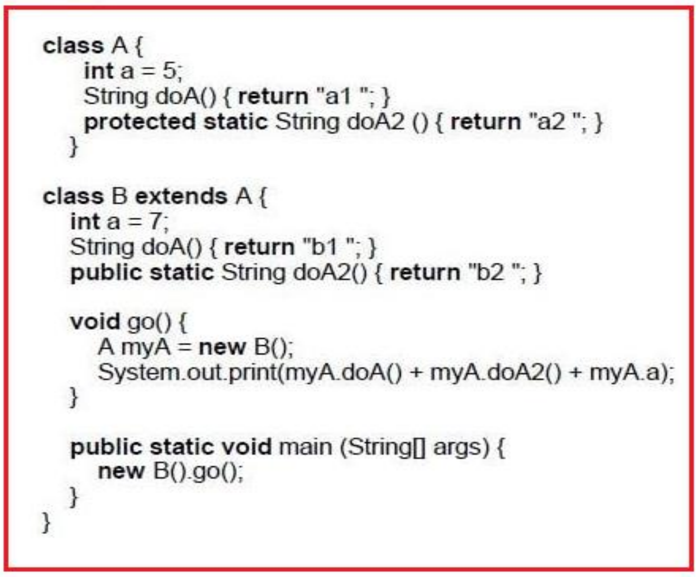
[1 mark]

```
import java.util.*;
public class q5 {
    public static void main(String[] args) {
```

```
int apu[] = {2, 4, 6, 8, 10, 12, 14, 16, 18, 20};
 int input, temp;
 Scanner scanner = new Scanner(System.in);
 for (int i=0; i < 5; i++) {
 input = scanner.nextInt();
 for (int j=0; j < apu.length; j++){
 if (input == apu[j]) {
        temp = apu[j];
 for (int k=j; k > 0; k--)
        apu[k] = apu[k-1];
 apu[0] = temp;
  }
 }
         }
 // output
 for (int j=0; j < apu.length; j++)
        System.out.print(apu[j]+ " ");
         }
 }
```

```
a)  12 6 2 4 8 10 14 16 18 20
b)  12 2 4 8 10 14 16 18 20 6
c)  12 6 2 8 10 14 16 18 4 20
d)  12 10 14 16 18 20 6 2 20
e)  12 6 2 4 8 10 14 16 14 20
```

Q6 What will be the output of the following code ?



```
class A {
   int a = 5;
   String doA() { return "a1 "; }
   protected static String doA2 () { return "a2 "; }
  }

class B extends A {
  int a = 7;
  String doA() { return "b1 "; }
  public static String doA2() { return "b2 "; }

  void go() {
    A myA = new B();
    System.out.print(myA.doA() + myA.doA2() + myA.a);
  }

  public static void main (String[] args) {
    new B().go();
  }
}
```

```
a)  b1a25
b)  b2a15
c)  a1b17
d)  b1a27
e)  a2b15
```

Q7 Consider the following headers of method:

```
(i) public int apuMethod(int a, char b)
(ii) public void apuMethod()
(iii) public int apuMethod(int a)
(iv) public int apuMethod(char b, int a)
(v) public int apuMethod(int b, char a)
(vi) public void apuMethod(int a, char b)
(vii) public void apuMethod(int a, char b, float c)
```

Which of the following could appear together in the same class?

```
a)  All seven can be used together.
b)  All except (i) can be used together.
c)  All except (v) can be used together.
d)  All except (vi) can be used together.
e)  All except (i) and (vi) can be used together.
```

Q8 Assume that the following three classes have been defined:

```
class JayZ extends Tupac {
    public void a() {
    System.out.print("JayZ a   ");
             }
}
```

```
class Biggie extends JayZ {
    public void a() {
     System.out.print("Biggie a   ");
        super.a();
    }

    public String toString() {
        return "Biggie";
    }
}
```

```
class Tupac {
    public void a() {
        System.out.print("Tupac a   ");
    }

    public void b() {
        System.out.print("Tupac b   ");
    }

    public String toString() {
        return "Tupac";
    }
}
```

Given the classes above, what output is produced by the following code?

```
Tupac[] elements = { new Tupac(), new JayZ(), new Biggie()};

   for (int i = 0; i < elements.length; i++)
   {
       System.out.println(elements[i]);
   }
```

```
a)  Tupac JayZ a Biggie a Jayz
b)  Tupac a   JayZ a   Biggie a   JayZ a
c)  Tupac a   JayZ a   Biggie  JayZ a
d)  Tupac b   JayZ a   Biggie a   JayZ a
e)  Tupac JayZ a Biggie Jayz a
```

Q9 What will happen when you compile and run the following code with class named Main?

```
public class Main{
               private int i = 1;
               public static void main(String argv[]){
               int i = 2;
               Main s = new Main ();
               s.someMethod();
               }
               public void someMethod(){
               System.out.println(i);
               }
               }

                   a)  2
                   b)  2 1
                   c)  1
                   d)  1 2
                   e)  None
```

Q10 What will happen when you compile and run the following code?

```
        import java.util.*;
        class Penisular {
        int lat;
        int lng;
        public Penisular(int initialLat, int initialLng) {
        lat = initialLat;
        lng = initialLng;
        }
        }

        public class q11 {

          public static void main(String[] args) {
               int n = 4;
        int[] a = {8, 15};
        Penisular isle = new Penisular(16, 23);
        n++;
        a[0] += 2;
        isle.lat = 42;
        mystery(n, a, isle);
            }
            public static void mystery(int n, int[] a, Penisular isle) {
        n = n - 2;
        a[1]++;
        isle.lng += 100;
   System.out.println(n + " " + Arrays.toString(a) + " " + isle.lat + " " + isle.lng);
                       }

                       }

                   a)  3 [11, 16] 42 123
                   b)  3 [10, 17] 42 123
                   c)  3 [9, 16] 42 123
                   d)  3 [10, 16] 42 123
                   e)  3 [10, 16] 41 123
```

Q11 Indicate the false statement from the list of statements given below:

a) In the UML, each class is modeled in a class diagram with three compartments. The top one contains the class's name centered horizontally in boldface. The middle one contains the class's attributes, which correspond to instance variables in Java. The bottom one contains the class's operations, which correspond to methods and constructors in Java.
b) UML represents classes using the box / rectangle shape.
c) Private attributes are preceded by the keyword private in the UML.
d) The UML models operations by listing the operation name followed by a set of parentheses. A plus sign (+) in front of the operation name indicates that the operation is a public.
e) none of the above

Q12 Indicate the false statement in the given list of statements about abstract classes

a) A class containing abstract methods is called an abstract class.
b) Abstract methods should be implemented in the derived class.
c) An abstract class cannot have non-abstract methods.
d) A class must be qualified as 'abstract' class, if it contains one abstract method.

e) None of the above.

Q13 Indicate the incorrect statement about the constructors in the given list.

a) It is possible to have many constructors, but not many methods in a class.
b) Constructors must have the same name as the class, but methods cannot have a class name.
c) Constructor initializes instance of the class whereas methods performs operations pertaining to the class itself.
d) Constructor is called whenever an object is created using new keyword, whereas methods will need to be invoked.
e) None of the above.