Question 1 out of 40 questions

Consider the following method.

```
// precondition: numArray contains no duplicates and the
// elements in numArray are in ascending sorted order and
// 0 <= low <= numArray.length and low - 1 <= high <
numArray.length
public static int mystery(int[] numArray, int low, int high, int
value)
{
   if (low > high)
      return low;

   int mid = (low + high) / 2;

   if (numArray[mid] == value)
      return mid;
   else if (numArray[mid] < value)
      return mystery(numArray, mid + 1, high, value);
   else
      return mystery(numArray, low, mid - 1, value);
}
```

How many calls to mystery (including the initial call) are made as a result of the call `mystery(numArray, 0, numArray.length - 1, 32);` if numArray is declared as follows?

```
int[] numArray = {2, 10, 23, 31, 35, 48, 69, 98};
```

○ Answer a:
3

○ Answer b:
4

○ Answer c:
8

○ Answer d:
1

○ Answer e:
2

Question 2 out of 40 questions

Which of the following statements about interfaces is (are) true?

```
I.  One interface can inherit from another.
II.  You can declare a abstract method in an interface.
III.  You can create a new object using an interface.  If I have
an
interface named List I can do List aList = new List();
```

□ Answer a:
II only

□ Answer b:
I only

□ Answer c:
III only

□ Answer d:
I and II only

□ Answer e:
I, II, and III

Question 3 out of 40 questions

Consider the following methods:

```
public List process1(int n)
{
  List<Integer> someList = new ArrayList<Integer>();
  for (int k = 0; k < n; k++)
    someList.add(k);
  return someList;
}
public List process2(int n)
{
  List<Integer> someList = new ArrayList<Integer>();
  for (int k = 0; k < n; k++)
    someList.add(k, k);
  return someList;
}
```

Which of the following best describes the behavior of process1 and process2?

□ Answer a:
Both methods produce the same result, and process1 is faster than process2.

□ Answer b:
Both methods produce the same result and take the same amount of time.

□ Answer c:
The two methods produce different results and take the same amount of time.

□ Answer d:
The two methods produce different results, and process2 is faster than process1.

□ Answer e:
The two methods produce different results, and process1 is faster than process2.

Question 4 out of 40 questions

Consider the following segment of code:

```
public static void modulusDivision(int n)
{
   if (n % 10 == 1)
   {
      System.out.println("End");
   }
   else
   {
      System.out.println("Recursing");
      modulusDivision(n/10);
   }
}
```

What will printed as a result of the call `modulusDivision(1001);`?

□ Answer a:

```
Recursing
Recursing
End
```

□ Answer b:

```
Recursing
Recursing
Recursing
Recursing
End
```

○ Answer c:

```
Recursing
End
```

○ Answer d:
End

○ Answer e:

```
Recursing
Recursing
Recursing
```

Question 5 out of 40 questions

Which of the following reasons for using an inheritance heirarchy are valid?

```
I.   Methods from a superclass can be used in a subclass without
     rewriting or copying code.
II.  Objects from subclasses can be passed as arguments to a
method
     designed for the superclass
III. Objects from subclasses can be stored in the same array
IV.  All of the above
V.   None of the above
```

○ Answer a:
IV

○ Answer b:
I only

○ Answer c:
V.

○ Answer d:
I and III

○ Answer e:
I and II

Question 6 out of 40 questions

Consider the following code segment:

```
for (int k = 0; k < 20; k = k + 2)
{
   if (k % 3 == 1)
      System.out.print(k + " ");
}
```

What is printed as a result of executing the code segment?

○ Answer a:
0 6 12 18

○ Answer b:
0 2 4 6 8 10 12 14 16 18

○ Answer c:
4 10 16

○ Answer d:
4 16

○ Answer e:
1 4 7 10 13 16 19

Question 7 out of 40 questions

What is the best explanation for what is meant by overriding a method?

○ Answer a:
Defining a method with the same name as an inherited method but with a different number of parameters

○ Answer b:
Defining another method with the same name as another method but with different types for the parameters

○ Answer c:
Defining another method with the same name as another method but with a different number of parameters

○ Answer d:
Defining a method with the same precondition as another method

□ Answer e:
Defining a method with the same name and parameter list as an inherited method

Question 8 out of 40 questions

Which of the following statements about a class that contains an abstract method is (are) true?

```
I. You can't have any constructors in this class.
II.  This class must be declared as abstract.
III.  You can't declare any fields in this class.
```

□ Answer a:
II only

□ Answer b:
I, II and III

□ Answer c:
I only

□ Answer d:
III only

□ Answer e:
I and II only

Question 9 out of 40 questions

Consider the following code segment:

```
if ( x > 0) x = -x;
if (x < 0) x = 0;
```

Which of the following is this equivalent to?

□ Answer a:

```
if (x > 0) x = -x;
else x = 0;
```

□ Answer b:

```
if ( x < 0) x = 0;
else x = -1;
```

□ Answer c:

```
if (x > 0) x = 0;
```

□ Answer d:

```
x = 0;
```

□ Answer e:

```
if (x < 0) x = 0;
```

Question 10 out of 40 questions

Which of the following is a reason to use a list (assume an object of the class ArrayList) instead of an array?

□ Answer a:
A list can store objects, but arrays can only store primitive types.

□ Answer b:
You can put a value in an array for an index but you can't do this with a list.

□ Answer c:
A list resizes itself as necessary as items are added, but an array does not.

□ Answer d:
A list has faster access to the last element than an array

□ Answer e:
A list uses less memory than an array

Question 11 out of 40 questions

Consider the following two classes.

```
public class Base
{
  public void methodOne()
  {
    System.out.print("A");
    methodTwo();
  }
  public void methodTwo()
  {
    System.out.print("B");
  }
}

public class Derived extends Base
{
  public void methodOne()
  {
    super.methodOne();
    System.out.print("C");
  }

  public void methodTwo()
  {
    super.methodTwo();
    System.out.print("D");
  }
}
```

Assume that the following declaration appears in a client program.

```
 Base b = new Derived();
```

What is the result of the call b.methodOne()?

○ Answer a:
ABC

○ Answer b:
AB

○ Answer c:
Nothing is printed due to infinite recursion.

○ Answer d:
ABDC

○ Answer e:
ABCD

Question 12 out of 40 questions

Consider the following code segment:

```
public static void stringMagic(String name)
{
   if(name.length() == 1)
   {
      System.out.println(name);
      return;
   }
   else
   {
      System.out.print(name.substring(name.length()-
1,name.length()));
      stringMagic(name.substring(0,name.length()-1));
   }
}
```

What is printed out if we call `stringMagic("Java Is Fun");`?

○ Answer a:
Java Is Fun

○ Answer b:
avaJ

○ Answer c:
nuF sI avaJ

○ Answer d:
nuFsIavaJ

○ Answer e:
Error: This is an infinite loop which will never terminate.

Question 13 out of 40 questions

Consider the following code segment:

```
List list1 = new ArrayList();
list1.add(new Integer(1));
list1.add(new Integer(2));
list1.add(new Integer(3));
list1.set(2, new Integer(4));
list1.add(2, new Integer(5));
list1.add(new Integer(6));
System.out.println(list1);
```

What is printed as a result of executing the code segment?

□ Answer a:
[1, 2, 5, 4, 6]

□ Answer b:
[1, 5, 2, 4, 6]

□ Answer c:
[1, 2, 4, 5, 6]

□ Answer d:
[1, 2, 3, 4, 5]

□ Answer e:
You get a compile error because of the .

Question 14 out of 40 questions

What all gets printed when the following method is run?

```
   public static void test() {
      for(int i = 0; i < 2; i++) {
         for(int j = 2; j>= 0; j--) {
            if(i != j) {
              System.out.println("i=" + i + " j="+j);
            }
         }
      }
   }
```

□ Answer a:

```
i=0 j=1
i=0 j=2
i=1 j=0
i=1 j=2
i=2 j=0
i=2 j=1
```

□ Answer b:

```
i=0 j=1
i=0 j=2
i=1 j=0
i=1 j=2
```

○ Answer c:

```
i=0 j=2
i=0 j=1
i=1 j=2
i=1 j=0
```

○ Answer d:

```
i=0 j=2
i=0 j=1
i=1 j=2
i=1 j=0
i=2 j=1
i=2 j=0
```

○ Answer e:

```
i=0 j=2
i=0 j=1
i=0 j=0
i=1 j=2
i=1 j=1
i=1 j=0
```

Question 15 out of 40 questions

Which of the following statements about abstract classes and interfaces is TRUE?

○ Answer a:
An abstract class cannot implement an interface

○ Answer b:
All subclasses of a class that implements an interface do not automatically implement the interface.

○ Answer c:
You can't declare variables with the type of an abstract class.

○ Answer d:
An abstract class cannot extend another abstract class

□ Answer e:
If an abstract class has no implemented constructors or methods, it is better to make it an interface.

Question 16 out of 40 questions

Assume that temp is an int variable initialized to be greater than zero and that a is an array of ints. Consider the following code segment:

```
   for ( int k = 0; k < a.length; k++ )
   {
      while ( a[ k ] < temp )
      {
         a[ k ] *= 2;
      }
   }
```

What of the following will cause an infinite loop?

□ Answer a:
Whenever a includes a value equal to temp.

□ Answer b:
Whenever a has values larger then temp.

□ Answer c:
Whenever a includes a value that is less than or equal to zero.

□ Answer d:
When all values in a are larger than temp.

□ Answer e:
The values don't matter this will always cause an infinite loop.

Question 17 out of 40 questions

Consider the following method.

```
 /**
   * Method to do a selection sort on
   * an array of integers
   */
  public static void selectionSort(int[] numberArray)
  {
    int temp; // used to hold value
    for (int i = 0; i < numberArray.length - 1; i++)
// line 1
    {
      int pos = i;
// line 2
      for (int j = 0; j < numberArray.length; j++)
// line 3
      {
        if (numberArray[j] < numberArray[pos])
// line 4
        {
          pos = j;
// line 5
        }
      }
      temp = numberArray[i];
      numberArray[i] = numberArray[pos];
      numberArray[pos] = temp;
    }
  }
```

This method should sort the numbers in the passed array into ascending order. But, it does not work. Which of the following changes would fix it?

□ Answer a:
Line 4 should be:

```
if (numberArray[j] > numberArray[pos])
```

□ Answer b:
Line 5 should be:

```
j = pos;
```

□ Answer c:
Line 3 should be:

```
for (int j = i + 1; j < numberArray.length; j++)
```

□ Answer d:
Line 1 should be:

```
for (int i = 0; i < numberArray.length - 2; i++)
```

□ Answer e:
Line 2 should be:

```
int pos = 0;
```

Question 18 out of 40 questions

A program is being written by a team of programmers. One programmer is implementing a class called Employee; another programmer is writing code that will use the Employee class. Which of the following aspects of the public methods and fields of the Employee class does **not** need to be known by both programmers?

□ Answer a:
Constants

□ Answer b:
The method names.

□ Answer c:
The method return types.

□ Answer d:
The number and types of the method parameters.

□ Answer e:
How the methods are implemented.

Question 19 out of 40 questions

Given the following declaration of a field in a class:

```
public static final String GREETING = "Hi";
```

Which of these statements is **not** true?

□ Answer a:
The value of GREETING can not be changed in any methods

□ Answer b:
Each object of this class has a copy of GREETING

□ Answer c:
GREETING.length() == 2

○ Answer d:
Each object of this class can access GREETING

○ Answer e:
GREETING.toUpperCase().equals("HI")

Question 20 out of 40 questions

Consider the following classes.

```
public abstract class Animal
{
   public void run()
   {
      System.out.println("Running");
   }
}
public class Cheetah extends Animal
{
   public void run()
   {
      super.run();
      System.out.println("Running really fast");
   }
}
```

What will be printed out when the below code segment is run?

```
Cheetah c = new Animal();
c.run();
```

○ Answer a:

```
Running really fast
Running
```

○ Answer b:

```
Running really fast
```

○ Answer c:

```
Running
Running really fast
```

○ Answer d:
Nothing will be printed, because of a compiler error.

○ Answer e:

```
Running
```

Question 21 out of 40 questions

Assume that variable b is an array of k integers and that the following is true:

```
b[0] != b[k] for all k such that 1 <= k
```

Which of the following statements is a valid conclusion?

○ Answer a:
Array b is not sorted

○ Answer b:
Array b is sorted

○ Answer c:
Array b contains no duplicates

○ Answer d:
The value in b[0] does not occur anywhere else in the array

○ Answer e:
The value in b[0] is the smallest value in the array

Question 22 out of 40 questions

Consider the following class declaration.

```
public class ParkingPass
{
   private int numLeft;

   // constructor not shown

   public int getNumLeft()
   {
      return numLeft;
   }
   public String toString()
   {
      return "Parking pass has " + numLeft + " left";
   }
}
```

If the following declaration appears in a client class.

```
ParkingPass pass = new ParkingPass();
```

Which of these statements can be used in the client class?

I. System.out.println(pass.numLeft);
II. System.out.println(pass);
III. System.out.println(pass.getNumLeft());

□ Answer a:
II only

□ Answer b:
I and III

□ Answer c:
II and III

□ Answer d:
III only

□ Answer e:
I only

Question 23 out of 40 questions

Consider the below code segment:

```
boolean temp = false;
for ( int i = 0; i < a.length; i++) {
   temp = ( a[i] == val );
}
return temp;
```

Which of the following statements best describes the conditions needed for temp = true?

□ Answer a:
Whenever exactly 1 element in a is equal to val.

○ Answer b:
Whenever the last element in a is equal to val.

○ Answer c:
Whenever the first element in a is equal to val

○ Answer d:
Whenever more than 1 element in a is equal to val.

○ Answer e:
Whenever a contains any element which equals val.

Question 24 out of 40 questions

Consider the following partial class declaration.

```
public class Person implements Comparable
{
   private String firstName;
   private String lastName;

   public int compareTo(Object test)
   {
      // implementation not shown
   }

   // constructors, other fields, and methods not shown
}
```

Assume that the Person objects are ordered by last name and then first name. Which of the following will correctly implement compareTo for the Person class?

```
I.  public int compareTo(Object test)
    {
       Person testP = (Person) test;
       return (lastName.compareTo(testP.lastName) +
               firstName.compareTo(testP.firstName));
    }

II. public int compareTo(Object test)
    {
       Person testP = (Person) test;
       if (firstName.compareTo(testP.firstName == 0)
          return lastName.compareTo(testP.lastName);
       else
          return firstName.compareTo(testP.firstName);
     }
```

```
III. public int compareTo(Object test)
     {
       Person testP = (Person) test;
       if (lastName.compareTo(testP.lastName == 0)
          return firstName.compareTo(testP.firstName);
       else
          return lastName.compareTo(testP.lastName);
     }
```

○ Answer a:
I and III only

○ Answer b:
III only

○ Answer c:
I only

○ Answer d:
II only

○ Answer e:
I and II only

Question 25 out of 40 questions

Which will cause the **longest** execution of a sequential search looking for a value in an array of 10 integers?

○ Answer a:
The value is at position 6 in the array

○ Answer b:
The value is in the middle of the array

○ Answer c:
The value is at position 3

○ Answer d:
The value is not in the array

□ Answer e:
The value is the first in the array

Question 26 out of 40 questions

You are given the following array of integers a[2,5,3,7,4,9,10,1]. What will be the final array after you execute the method mystery using this array?

```
public static void mystery(int[] a){
   for(int i = 1; i < a.length; i++)
   {
      int current = a[i];
      int j = i-1;

      while((j >= 0) && (a[j] > current)){
         a[j+1] = a[j--];
      }

      a[j+1] = current;
   }
}
```

□ Answer a:
a[2,1,3,4,5,7,9,10]

□ Answer b:
a[2,5,3,7,4,9,10,1]

□ Answer c:
a[1,2,5,3,7,4,9,10]

□ Answer d:
a[10,9,7,5,4,3,2,1]

□ Answer e:
a[1,2,3,4,5,7,9,10]

Question 27 out of 40 questions

The following method attempts to perform an insertion sort:

```
      public void sort()
0:    {
```

```
1:       for (int i = 1; i < a.length; i++)
2:       {
3:          int next = a[i];

4:          // Move all larger elements to the right
5:          int j = i;
6:          while (j > 0 && a[j - 1] > next)
7:          {
8:             a[j-1] = a[j];
9:             j--;
10:          }

11:          // Insert the element
12:          a[j] = next;
13:       }
14:    }
```

However, it does not work properly. Which is the line that contains an error?

□ Answer a:
In line 8 it should be `a[j] = a[j-1];`

□ Answer b:
In line 1 the code `i < a.length` should be `i < a.length ? 1`.

□ Answer c:
In line 6 the code `j > 0` should be `j < i`.

□ Answer d:
In line 5, should be int j = i + 1;

□ Answer e:
In line 1 the code should be `for (int i = 0; i < a.length; i++)`

Question 28 out of 40 questions

If you have the following classes:

```
public class Point2D {
  public int x;
  public int y;

  public Point2D() {}
  public Point2D(int x,int y) {
     this.x = x;
     this.y = y;
  }
  // other methods
}

public class Point3D extends Point2D
{
   public int z;

   // other code
}
```

Which of the following constructors would be valid for Point3D?

```
I.  public Point3D() {}
II. public Point3D(int x, int y, int z)
    {
       super(x,y);
       this.z = z;
    }
III. public Point3D(int x, int y)
     {
        this.x = x;
        this.y = y;
        this.z = 0;
     }
```

○ Answer a:
III only

○ Answer b:
I, II, and III

○ Answer c:
II only

○ Answer d:
I only

○ Answer e:
I and II only

Question 29 out of 40 questions

Given the following class definitions:

```
public class ContactInfo
{
  private String name;
  private String phoneNumber;

  public ContactInfo(String theName, String thePhoneNumber)
  {
    this.name = theName;
    this.phoneNumber = thePhoneNumber;
  }

  public String getName() { return name; }

  public String getPhoneNumber() { return phoneNumber; }
}

public class ExtendedContactInfo extends ContactInfo
{
  private String nickname;

  public ExtendedContactInfo (String theNickname,
                              String theName,
                              String thePhoneNumber)
  {
    // missing code
  }
}
```

Which of the following can replace the // missing code?

○ Answer a:

```
this.nickname = theNickname;
super(theName, thePhoneNumber);
```

○ Answer b:

```
this.nickname = theNickname;
this.name = theName;
this.phoneNumber = thePhoneNumber;
```

○ Answer c:

```
this.name = theName;
this.phoneNumber = thePhoneNumber;
this.nickname = theNickname;
```

□ Answer d:

```
super(theNickname, theName, thePhoneNumber);
```

□ Answer e:

```
super(theName,thePhoneNumber);
this.nickname = theNickname;
```

Question 30 out of 40 questions

Consider the following data field and method:

```
private List<Integer> nums;

// precondition: nums.size() > 0;
// nums contains Integer objects
public void numQuest()
{
  int k = 0;
  Integer zero = new Integer(0);
  while (k < nums.size())
  {
    if (nums.get(k).equals(zero))
      nums.remove(k);
    k++;
  }
}
```

Assume that nums has been created as an ArrayList object and it initially contains the following Integer values. [0, 0, 4, 2, 5, 0, 3, 0] What will nums contain as a result of executing numQuest ?

□ Answer a:
[0, 0, 4, 2, 5, 0, 3, 0]

□ Answer b:
[0, 4, 2, 5, 3]

□ Answer c:
[0, 0, 0, 0, 4, 2, 5, 3]

○ Answer d:
[3, 5, 2, 4, 0, 0, 0, 0]

○ Answer e:
[4, 2, 5, 3]

Question 31 out of 40 questions

```
public class Person implements Comparable
{ code for class including a compareTo() method }
public class Player extends Person
{ code for class }
```

Which declaration will result in a compiler error?

○ Answer a:
Person p = new Person();

○ Answer b:
Comparable c = new Person();

○ Answer c:
Person p = new Player();

○ Answer d:
Player p = new Person();

○ Answer e:
Comparable c = new Player();

Question 32 out of 40 questions

Assume that x is an initialized int variable. The code segment:

```
if (x > 2) x = x * 2;
if (x > 4) x = 0;
```

is equivalent to which of the following code segments?

○ Answer a:

```
if (x > 2) x *= 2;
else if (x > 4) x = 0;
```

○ Answer b:

```
if (x > 2) x = 0;
```

○ Answer c:

```
x = 0;
```

○ Answer d:

```
if (x > 2) x *= 2;
```

○ Answer e:

```
if (x > 2) x = 0;
else x *= 2;
```

Question 33 out of 40 questions

Consider the following data field and method findLongest with line numbers added for reference. Method findLongest is intended to find the longest consecutive block of the value target occurring in the array nums; however, findLongest does not work as intended. For example, if the array nums contains the values [7, 10, 10, 15, 15, 15, 15, 10, 10, 10, 15, 10, 10], the call findLongest(10) should return 3, the length of the longest consecutive block of 10s.

```
      private int[] nums;
      public int findLongest(int target)
      {
        int lenCount = 0;
        int maxLen = 0;

Line 1: for (int k = 0; k < nums.length; k++)
Line 2: {
Line 3:   if (nums[k] == target)
Line 4:   {
Line 5:     lenCount++;
Line 6:   }
Line 7:   else
Line 8:   {
Line 9:     if (lenCount > maxLen)
Line 10:    {
Line 11:      maxLen = lenCount;
Line 12:    }
Line 13:  }
Line 14: }
Line 15: if (lenCount > maxLen)
Line 16: {
```

```
Line 17:   maxLen = lenCount;
Line 18: }
Line 19: return maxLen;
       }
```

The method findLongest does not work as intended. Which of the following best describes the value returned by a call to findLongest ?

□ Answer a:
It is the length of the last consecutive block of the value target in nums.

□ Answer b:
It is the number of occurrences of the value target in nums.

□ Answer c:
It is the length of the array nums.

□ Answer d:
It is the length of the shortest consecutive block of the value target in nums.

□ Answer e:
It is the length of the first consecutive block of the value target in nums.

Question 34 out of 40 questions

Given the following method:

```
public int mystery(int n)
{
  if (n == 0) return 1;
  else return 2 *
         mystery (n - 1);
}
```

What value is returned as the result of mystery(7)?

□ Answer a:
2

□ Answer b:
256

□ Answer c:
64

□ Answer d:
0

□ Answer e:
128

Question 35 out of 40 questions

Array unsortedArr contains an unsorted list of integers. Array sortedArr contains a sorted list of integers. Which of the following operations is more efficient for sortedArr then unsortedArr? Assume the most efficient algorithms are used.

I   Inserting a new element
II Searching for a given element
III Computing the mean of the elements

(a) I only
(b) II only
(c) III only
(d) I and II only
(e) I, II and III

□

Answer a:

(e)

□

Answer b:

(b)

○

Answer c:

(d)

○

Answer d:

(a)

○

Answer e:

(c)

Question 36 out of 40 questions

Which of the following scenarios correctly shows the iterations of an ascending (from left to right) insertion sort on an array with the following elements:
{7,8,5,3}

○ Answer a:
None of these answers are correct

○ Answer b:

{7,8,5,3}
{7,5,8,3}

```
{7,5,3,8}
{5,7,3,8}
{5,3,7,8}
{3,5,7,8}
```

□ Answer c:

```
{7,8,5,3}
{5,7,8,3}
{3,5,7,8}
```

□ Answer d:

```
{3,8,5,7}
{3,5,8,7}
{3,5,7,8}
```

□ Answer e:

```
{7,8,5,3}
{7,8}{5,3}
{7}{8}{5}{3}
{7,8}{3,5}
{3,5,7,8}
```

Question 37 out of 40 questions

Consider the following:

```
String s1 = "Hi There";
String s2 = s1;
String s3 = s2;
String s4 = s1;
s2 = s2.toLowerCase();
s3 = s3.toUpperCase();
s4 = null;
```

What string is referrenced by s1?

□ Answer a:
Hi There

□ Answer b:
null

□ Answer c:
HI THERE

□ Answer d:
hi there

□ Answer e:
hI tHERE

Question 38 out of 40 questions

Consider the following declaration for a class that will be used to represent points in the xy-coordinate plane:

```
public class Point
{
  private int myX; // coordinates
  private int myY;

  public Point( )
  {
    myX = 0;
    myY = 0;
  }

  public Point(int a, int b)
  {
    myX = a;
    myY = b;
  }

// ... other methods not shown

}
```

The following incomplete class declaration is intended to extend the above class so that two-dimensional points can be named.

```
public class NamedPoint extends Point
{
  private String myName;
  // constructors go here
  // ... other methods not shown
}
```

Consider the following proposed constructors for this class:

```
I.   public NamedPoint()
     {
       myName = "";
     }
II.  public NamedPoint(int d1, int d2, String name)
     {
       myX = d1;
       myY = d2;
       myName = name;
     }
III. public NamedPoint(int d1, int d2, String name)
     {
       super(d1, d2);
       myName = name;
     }
```

Which of these constructors would be legal for the NamedPoint class?

○ Answer a:
I only

○ Answer b:
II only

○ Answer c:
III only

○ Answer d:
I and III

○ Answer e:
I and II

Question 39 out of 40 questions

What will be printed out at the end of this code snippet?

```
public abstract class Fruit
{
   public void eat()
   {
      System.out.println("Eating fruit");
   }
}

public class Banana extends Fruit
{
   public void eat()
   {
      System.out.println("Eating banana");
   }
}

public class Mango extends Fruit
{
   public void eat()
   {
      super.eat();
      System.out.println("Eating mango");
   }
}

public static void main (String[] args)
{
   Fruit f = new Mango();
   f.eat();
}
```

○ Answer a:

```
Eating fruit
Eating mango
```

○ Answer b:

```
Eating fruit
```

○ Answer c:
Compiler error

○ Answer d:

```
Eating mango
```

○ Answer e:

```
Eating banana
```

Question 40 out of 40 questions

Class C extends class B, which extends class A. Also, all of the three classes implement a public method test(). How can a method in an object of class C invoke the test() method defined in class A (without creating a new instance of class A)?

□ Answer a:
super.test()

□ Answer b:
test();

□ Answer c:
this.test()

□ Answer d:
super.super.test();

□ Answer e:
It is not possible to invoke the test() method defined in class A from a method in class C.