Question 1 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 isn't in the array

Answer c:
The value is in the middle of the array

Answer d:
The value is the first one in the array

Answer e:
The value is at position 3 in the array

Question 2 out of 40 questions

Given the following method:

```
public boolean check(String s)
{
  return s.length() >= 2 && (s.charAt(0) ==
       s.charAt(1) || check(s.substring(1)));
}
```

This method will return true if and only if:

Answer a:
s contains two or more of the same character in a row

Answer b:
s.charAt(0) == s.charAt(1)

Answer c:
s contains two or more of the same characters

Answer d:
s ends with two or more of the same characters

Answer e:
s starts with two or more of the same characters

Question 3 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:
I and II only

Answer c:
I only

Answer d:
II only

Answer e:
III only

Question 4 out of 40 questions

Which of the following best describes the purpose of a methods pre and post conditions?

Answer a:
They communicate with the compiler

Answer b:
They permit the method to be in a different file than the code which calls the method

Answer c:
They initialize necessary variables so the method will run correctly.

Answer d:
They provide information to the programmer or reader about what the method is intended to do.

Answer e:
They explain how a method was implemented

Question 6 out of 40 questions

A track team needs to print a list of the practice times for each runner in the 100-meter dash. There are three runners, and each runner is measured four times. Given the following declaration:

```
double[][] times ={
```

```
    {11.98, 12.02, 12.05, 12.03},
    {11.85, 11.88, 11.86, 11.91},
    {12.04, 12.07, 12.11, 12.13}};
```

Which code segment will produce the following output:

```
Runner 1: 11.98  12.02  12.05  12.03
Runner 2: 11.85  11.88  11.86  11.91
Runner 3: 12.04  12.07  12.11  12.13
```

Answer a:

```
for (int runner=0; runner < times.length; runner++)
{
      System.out.print("Runner " + runner + 1 + ": ");
      for (int race=0; race < times[0].length; race++)
      {
        System.out.print(times[runner][race] + "  ");
      }
}
      System.out.println();
```

Answer b:

```
for (int race=0; race  < times.length; race++)
{
    System.out.print("Runner " + (race + 1) + ": ");
    for (int runner=0; runner < times[0].length; runner++)
    {
       System.out.print(times[runner][race] + "  ");
    }
    System.out.println();
}
```

Answer c:

```
for (int runner=0; runner < times.length; runner++)
{
    System.out.print("Runner " + (runner + 1) + ": ");
    for (int race=0; race < times[0].length; race++)
    {
        System.out.print(times[runner][race] + "  ");
    }
    System.out.println();
}
```

Answer d:

```
for (int runner=0; runner < times.length; runner++)
{
    System.out.print("Runner " + (runner + 1) + ": ");
    for (int race=0; race < times[0].length; race++)
     {
        System.out.print(times[race][runner] + "  ");
     }
    System.out.println();
}
```

Answer e:

```
for (int runner=0; runner < times[0].length; runner++)
{
    System.out.print("Runner " + (runner + 1) + ": ");
    for (int race=0; race < times.length; race++)
    {
        System.out.print(times[runner][race] + "  ");
    }
    System.out.println();
}
```

Question 7 out of 40 questions

If you have a parent class Animal that has a method speak() which returns: Awk and you have children classes that do the following:

- Cat has a speak method that returns: Meow
- Bird does not have a speak method
- Dog has a speak method that returns: Woof
- Pig does not have a speak method
- Cow has a speak method that returns: Moo

What is the output from looping through this array of animals starting at index 0 and asking each to speak()?

```
Animal[] a = { new Cat(), new Cow(), new Dog(), new Pig(), new
Bird() }
```

Answer a:
Meow Moo Woof Awk Awk

Answer b:
Meow Moo Woof Oink Awk

Answer c:
Awk Awk Awk Awk Awk

Answer d:
This will have runtime errors

Answer e:
This will not compile

Question 8 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:
avaJ

Answer b:
nuF sI avaJ

Answer c:
nuFsIavaJ

Answer d:
Java Is Fun

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

Question 9 out of 40 questions

Which one of the following statements about method overloading and overriding is true?

Answer a:
Overloading and overriding of methods are interchangeable terms in the object-oriented paradigm.

Answer b:
In overloading, two methods with the same name can have the same sequence of parameters as long as the parameter names are different.

Answer c:
Overriding means that two methods in the same class have the same name, but different parameter lists.

Answer d:
Overloading two methods means that one of the method names has to be different than the other.

Answer e:
Overriding allows for polymorphism which means that the actual method that gets called at runtime depends on the type of the object at runtime.

Question 10 out of 40 questions

Consider the following code segment:

```
char test[] = {'t', 'e', 's', 't'};
String str = new String("test");
```

Which of the following would not compile?

Answer a:

```
str.charAt(3)
```

Answer b:

```
int size = test.size();
```

Answer c:

```
String s = str.substring(2,3);
```

Answer d:

```
All of the code will compile
```

Answer e:

```
System.out.println(str);
```

Question 11 out of 40 questions

What is the output of the following code when compiled and run?

```
  public static void test()
  {
    int num = 0;
    while(num <= 14)
    {
      if(num % 3 == 1)
      {
        System.out.print("1 ");
      }
      else if (num % 3 == 2)
      {
        System.out.print("2 ");
      }
      else
      {
        System.out.print("0 ");
      }

      num += 2;
    }
}
```

Answer a:
0 2 1 0 2 1 0 2

Answer b:
0 2 1 0 2 1 0

Answer c:
2 1 0 2 1 0 2 1

Answer d:
0 1 2 0 1 2 0 1

Answer e:
0 2 1 0 2 1 0 2 1

Question 12 out of 40 questions

Consider the following code segment:

```
int [][] mat = new int [3][4];
for (int row = 0; row < mat.length; row++)
{
  for (int col = 0; col < mat[0].length; col++)
  {
    if (row < col)
      mat[row][col] = 1;
    else if (row == col)
      mat[row][col] = 2;
    else
      mat[row][col] = 3;
  }
}
```

What are the contents of mat after the code segment has been executed?

Answer a:

```
{{2 1 1},
 {3 2 1},
 {3 3 2},
 {3 3 3}}
```

Answer b:

```
{{2 3 3 3},
 {1 2 3 3},
 {1 1 2 3}}
```

Answer c:

```
{{2 3 3},
 {1 2 3},
 {1 1 2},
 {1 1 1}}
```

Answer d:

```
{{1 1 1 1},
 {2 2 2 2},
 {3 3 3 3}}
```

Answer e:

```
{{2 1 1 1},
 {3 2 1 1},
 {3 3 2 1}}
```

Question 13 out of 40 questions

Question: Consider the following method and code:

```
public int m1(int[] a){
   a[1]--;
   return (a[1] * 2);
}
// assume this code is in another method
int[] b = {2, 3, 4};
b[0] += m1(b);
for(int x: b)
   System.out.print(x+" ");
```

What is the run output of the code?

Answer a:
6 4 4

Answer b:
6 3 4

Answer c:
x x x

Answer d:
6 2 4

Answer e:
runtime error

Question 14 out of 40 questions

A two-dimensional array: imagePixels, holds the brightness values for the pixels in an image. The brightness can range from 0 to 255.

```
public int findMax(int[][] imagePixels)
{
     int r, c;
     int i, iMax = 0;

     for(r = 0; r < imagePixels.length; r++) {
         for(c = 0; c < imagePixels[0].length; c++) {
             i = image[r][c];
             if(i > iMax)
                iMax = i;
         }
      }
      return iMax;
}
```

What does this method compute?

Answer a:
The maximum brightness value for all pixels in imagePixels

Answer b:
The row with the greatest brightness sum

Answer c:
The most frequent brightness value in imagePixels

Answer d:
The column with the greatest brightness sum

Answer e:
The sum of the total brightness of imagePixels

Question 15 out of 40 questions

Consider the following code segment

```
for(int i = 0; i < 5; i++) {
   for(int j=0; j < 5; j++)
      System.out.println("*");
}
```

How many stars are output when this code is executed?

Answer a:
50

Answer b:
10

Answer c:
25

Answer d:
5

Answer e:
15

Question 16 out of 40 questions

Given the following incomplete class declaration:

```
public class TimeRecord
{
  private int hours;
  private int minutes; // 0<=minutes<60

  public TimeRecord(int h, int m)
  {
    hours = h;
    minutes = m;
  }

  // postcondition: returns the
  // number of hours
  public int getHours()
  { /* implementation not shown */ }

  // postcondition: returns the number
  // of minutes; 0 <= minutes < 60
  public int getMinutes()
  { /* implementation not shown */ }

  // precondition: h >= 0; m >= 0
  // postcondition: adds h hours and
  // m minutes to this TimeRecord
  public void advance(int h, int m)
  {
```

```
    hours = hours + h;
    minutes = minutes + m;
    /* missing code */
  }

  // ... other methods not shown
}
```

Consider the following declaration that appears in a client program:

```
TimeRecord[] timeCards = new TimeRecord[100];
```

Assume that timeCards has been initialized with TimeRecord objects. Consider the following code segment that is intended to compute the total of all the times stored in timeCards.

```
TimeRecord total = new TimeRecord(0,0);
for (int k = 0; k < timeCards.length; k++)
{
/* missing expression */ ;
}
```

Which of the following can be used to replace /* missing expression */ so that the code segment will work as intended?

Answer a:

```
timeCards[k].advance(timeCards[k].getHours(),
                                    timeCards[k].getMinutes())
```

Answer b:

```
total += timeCards[k].advance()
```

Answer c:

```
total.advance(timeCards[k].getHours(),
                      timeCards[k].getMinutes())
```

Answer d:

```
total.advance(timeCards[k].hours,
                     timeCards[k].minutes)
```

Answer e:

```
timeCards[k].advance()
```

Question 17 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 more than 1 element in a is equal to val.

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

Answer c:
Whenever a contains any element which equals val.

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

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

Question 18 out of 40 questions

A car dealership needs a program to store information about the cars for sale. For each car, they want to keep track of the following information: number of doors (2 or 4), whether the car has air conditioning, and its average number of miles per gallon. Which of the following is the best design?

Answer a:
Use a class Car which has three subclasses: Doors, AirConditioning, and MilesPerGallon.

Answer b:
Use one class, Car, which has three data fields: int numDoors, boolean hasAir, and double milesPerGallon.

Answer c:
Use three classes: Doors, AirConditioning, and MilesPerGallon, each with a subclass Car.

Answer d:
Use a class Car, which has a subclass Doors, with a subclass AirConditioning, with a subclass MilesPerGallon.

Answer e:
Use four unrelated classes: Car, Doors, AirConditioning, and MilesPerGallon.

Question 19 out of 40 questions

Which of the following statements assigns the letter S to the third row and first column of a two-dimensional array named letterGrid (assuming row-major order).

Answer a:
```
letterGrid[0][2] = "S";
```

Answer b:
```
letterGrid.setValue(2,0,"S");
```

Answer c:
```
letterGrid[2][0] = "S";
```

Answer d:
```
letterGrid[1][3] = "S";
```

Answer e:
```
letterGrid[3][1] = "S";
```

Question 20 out of 40 questions

Which of the following is the decimal value for the following binary number?
`1001011`

Answer a:
150

Answer b:
75

Answer c:
67

Answer d:
43

Answer e:
74

Question 21 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:

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

Answer c:

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

Answer d:

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

Answer e:

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

Question 22 out of 40 questions

What does the following method return when called with f(5)?

```
public static int f(int n)
  {
    if (n == 0)
      return 0;
    else if (n == 1)
      return 1;
    else return f(n-1) + f(n-2);
  }
```

Answer a:
5

Answer b:
There is no result because of infinite recursion.

Answer c:
0

Answer d:
8

Answer e:
3

Question 23 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[10,9,7,5,4,3,2,1]

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[2,1,3,4,5,7,9,10]

Answer e:
a[1,2,3,4,5,7,9,10]

Question 24 out of 40 questions

```
public class Student {
  private String getFood() {
     return "Pizza";
  }
```

```
  public String getInfo()  {
     return this.getFood();
  }
}

public class GradStudent extends Student {
  private String getFood() {
     return "Taco";
  }
  public void teach(){
     System.out.println("Education!");
  }
}
What is the output from this:

Student s1 = new GradStudent();
s1.getInfo();
```

Answer a:
Taco

Answer b:
Won't compile since you use this.getFood()

Answer c:
Won't compile since you are creating a GradStudent, not a Student

Answer d:
Won't compile since GradStudent doesn't have a getInfo method.

Answer e:
Pizza

Question 25 out of 40 questions

If you have the following:

```
String s1 = new String("Hi There");
String s2 = new String("Hi There");
String s3 = s1;
```

Which of the following would return true?

```
I.   (s1 == s2)
II.  (s1.equals(s2))
III. (s1 == s3)
IV.  (s2.equals(s3))
```

Answer a:
II, III and IV

Answer b:
II and IV

Answer c:
II only

Answer d:
I, II, III, IV

Answer e:
IV only

Question 26 out of 40 questions

Consider the following method:

```
public boolean checkID (int id, String name, Student nextStudent)
{
   if ((nextStudent.getName()).equals(name) && id ==
nextStudent.getID())
      return true;
   else
   {
      id = nextStudent.getID();
      nextStudent.setID(0);
      nextStudent.setName("VACANT");
      return false;
   }
}
```

Assume a class `Student` is defined with private instance fields `name` and `ID`. The class also has public accessors and mutators for these two fields. The method intends to check that the `name`and `ID` fields of the passed `nextStudent` match the passed `name` and `id`. If the names and/or the IDs do not match then it sets `id` to the value of `nextStudent.getID()`, resets the fields`ID` and `name`, and returns `false`. The intention is for the calling program to check for false on the return and then get the incorrect id stored in the variable that was passed as the parameter `id`, but this doesn't work. Which answer best describes why this doesn't work?

Answer a:
No instance field of a class can be changed by a client method if it is declared `private`.

Answer b:
The method will not compile since you use `(nextStudent.getName()).equals(name)`

Answer c:
A method can not pass an object as a parameter.

Answer d:
The method will not compile since there are two `return` statements in it.

Answer e:
If you modify a primitive type parameter in Java in a method it will not change the value of the variable in the calling method.

Question 27 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()
   {
      System.out.println("Running really fast");
   }
}
```

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

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

Answer a:

```
Running
Running really fast
```

Answer b:

```
Running really fast
```

```
Running
```

Answer c:
Nothing will be printed, there will be a runtime error.

Answer d:

```
Running really fast
```

Answer e:

```
Running
```

Question 28 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 and II

Answer b:
I and III

Answer c:
II only

Answer d:
I only

Answer e:
III only

Question 29 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 6 the code `j > 0` should be `j < i`.

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

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

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

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

Question 30 out of 40 questions

Given the following method:

```
public boolean check(String s)
{
  return s.length() >= 2 && (s.charAt(0) ==
       s.charAt(1) || check(s.substring(1)));
}
```

This method will return true if and only if:

Answer a:
s contains two or more of the same character in a row

Answer b:
s.charAt(0) == s.charAt(1)

Answer c:
s starts with two or more of the same characters

Answer d:
s contains two or more of the same characters

Answer e:
s ends with two or more of the same characters

Question 31 out of 40 questions

```
public class Student {
  public String getFood() {
     return "Pizza";
  }
  public String getInfo()  {
     return this.getFood();
  }
}

public class GradStudent extends Student {
  public String getFood() {
     return "Taco";
  }
}
```

What is the output from this:

```
Student s1 = new GradStudent();
s1.getInfo();
```

Answer a:
Won't compile since you are creating a GradStudent, not a Student

Answer b:
Won't compile since you use this.getFood()

Answer c:
Taco

Answer d:
Won't compile since GradStudent doesn?t have a getInfo method

Answer e:
Pizza

Question 32 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 2 4 6 8 10 12 14 16 18

Answer b:
1 4 7 10 13 16 19

Answer c:
4 16

Answer d:
4 10 16

Answer e:
0 6 12 18

Question 33 out of 40 questions

Consider the following field and method.

```
private int[] numArray;

public int getRunSize()
{
   if (numArray.length == 0)
```

```
      return 0;
   int size = 1;
   while ((size < numArray.length) &&
          (numArray[size - 1] > numArray[size]))
   {
      size++;
   }

   // assertion
   return size;
}
```

Which of the following assertions is true when execution reaches the line // assertion in getRunSize?

Answer a:
`(size == numArray.length) || (numArray[size - 1] <= numArray[size])`

Answer b:
`(size < numArray.length) && (numArray[size - 1] > numArray[size])`

Answer c:
`(size == numArray.length) && (numArray[size - 1] <= numArray[size])`

Answer d:
`(size < numArray.length) || (numArray[size - 1] > numArray[size])`

Answer e:
`(size == numArray.length) || (numArray[size - 1] == numArray[size])`

Question 34 out of 40 questions

Consider the following code segment:

```
if(!somethingIsTrue())
     return false;
else
     return true;
}
```

Which one of the following statements would be an accurate replacement for this code?

Answer a:
return !somethingIsTrue();

Answer b:
return somethingIsTrue();

Answer c:
return true;

Answer d:
None of these answers

Answer e:
return false;

Question 35 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:
I only

Answer b:
I, II and III

Answer c:
III only

Answer d:
I and II only

Answer e:
II only

Question 36 out of 40 questions

Consider the following declarations:

```
int valueOne = 3;
int valueTwo = 3;
```

Which of the following will compile and evaluate to true?

Answer a:
valueOne.equals((Integer) valueTwo)

Answer b:
valueOne == valueTwo

Answer c:
valueOne.compareTo(valueTwo) == 0

Answer d:
valueOne.equals(valueTwo)

Answer e:
valueOne.compareTwo((Integer) valueTwo) == 0

Question 37 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:
How the methods are implemented.

Answer b:
The method return types.

Answer c:
The method names.

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

Answer e:
Constants

Question 38 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
```

Answer b:

```
Running really fast
Running
```

Answer c:

```
Running
```

Answer d:

Nothing will be printed, because of a compiler error.

Answer e:

```
Running
Running really fast
```

Question 39 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:
Comparable c = new Player();

Answer b:
Player p = new Person();

Answer c:
Person p = new Person();

Answer d:
Person p = new Player();

Answer e:
Comparable c = new Person();

Question 40 out of 40 questions

Consider the following recursive method.

```
public static int mystery(int n)
{
  if (n == 0)
    return 1;
  else
    return 3 * mystery (n - 1);
}
```

What value is returned as the result of the call mystery(5)?

Answer a:
81

Answer b:
3

Answer c:
27

Answer d:
0

Answer e:
243