



# Chapter 11

## Practice Test 2

# AP<sup>®</sup> Computer Science A Exam

## SECTION I: Multiple-Choice Questions

**DO NOT OPEN THIS BOOKLET UNTIL YOU ARE TOLD TO DO SO.**

### At a Glance

**Total Time**

1 hour 30 minutes

**Number of Questions**

40

**Percent of Total Score**

50%

**Writing Instrument**

Pencil required

### Instructions

Section I of this examination contains 40 multiple-choice questions. Fill in only the ovals for numbers 1 through 40 on your answer sheet.

Indicate all of your answers to the multiple-choice questions on the answer sheet. No credit will be given for anything written in this exam booklet, but you may use the booklet for notes or scratch work. After you have decided which of the suggested answers is best, completely fill in the corresponding oval on the answer sheet. Give only one answer to each question. If you change an answer, be sure that the previous mark is erased completely. Here is a sample question and answer.

Sample QuestionSample Answer

Chicago is a

- (A) state
- (B) city
- (C) country
- (D) continent
- (E) county

(A) ☒ (C) (D) (E)

Use your time effectively, working as quickly as you can without losing accuracy. Do not spend too much time on any one question. Go on to other questions and come back to the ones you have not answered if you have time. It is not expected that everyone will know the answers to all the multiple-choice questions.

### About Guessing

Many candidates wonder whether or not to guess the answers to questions about which they are not certain. Multiple-choice scores are based on the number of questions answered correctly. Points are not deducted for incorrect answers, and no points are awarded for unanswered questions. Because points are not deducted for incorrect answers, you are encouraged to answer all multiple-choice questions. On any questions you do not know the answer to, you should eliminate as many choices as you can, and then select the best answer among the remaining choices.

**GO ON TO THE NEXT PAGE.**

## Quick Reference

### **class java.lang.Object**

- boolean equals(Object other)
- String toString()

### **class java.lang.Integer**

- Integer(int value)
- int intValue()
- Integer.MIN\_VALUE // minimum value represented by an int or Integer
- Integer.MAX\_VALUE // maximum value represented by an int or Integer

### **class java.lang.Double**

- Double(double value)
- double doubleValue()

### **class java.lang.String**

- int length()
- String substring(int from, int to) // returns the substring beginning at from  
// and ending at to-1
- String substring(int from) // returns substring(from, length())
- int indexOf(String str) // returns the index of the first occurrence of str;  
// returns -1 if not found
- int compareTo(String other) // returns a value < 0 if this is less than other  
// returns a value = 0 if this is equal to other  
// returns a value > 0 if this is greater than other

### **class java.lang.Math**

- static int abs(int x)
- static double abs(double x)
- static double pow(double base, double exponent)
- static double sqrt(double x)
- static double random() // returns a double in the range [0.0, 1.0)

### **interface java.util.List<E>**

- int size()
- boolean add(E obj) // appends obj to end of list; returns true
- void add(int index, E obj) // inserts obj at position index ( $0 \leq \text{index} \leq \text{size}$ ),  
// moving elements at position index and higher  
// to the right (adds 1 to their indices) and adjusts size
- E get(int index) // replaces the element at position index with obj
- E set(int index, E obj) // returns the element formerly at the specified position
- E remove(int index) // removes element from position index, moving elements  
// at position index + 1 and higher to the left  
// (subtracts 1 from their indices) and adjusts size  
// returns the element formerly at the specified position

### **class java.util.ArrayList<E> implements java.util.List<E>**

## COMPUTER SCIENCE A

## SECTION I

Time—1 hour and 30 minutes

Number of Questions—40

Percent of total exam grade—50%

**Directions:** Determine the answer to each of the following questions or incomplete statements, using the available space for any necessary scratchwork. Then decide which is the best of the choices given and fill in the corresponding oval on the answer sheet. No credit will be given for anything written in the examination booklet. Do not spend too much time on any one problem.

**Notes:**

- Assume that the classes listed in the Quick Reference have been imported where appropriate.
- Assume that declarations of variables and methods appear within the context of an enclosing class.
- Assume that method calls that are not prefixed with an object or class name and are not shown within a complete class definition appear within the context of an enclosing class.
- Unless otherwise noted in the question, assume that parameters in the method calls are not `null` and that methods are called only when their preconditions are satisfied.

## MULTIPLE CHOICE QUESTIONS

## USE THIS SPACE FOR SCRATCHWORK

1. Consider the following method:

```
public static int mystery(int a, int b)
{
    if (a <= 0)
        return b;
    else
        return mystery(a - 2, b);
}
```

What value is returned by the call `mystery (12, 5)`?

- (A) 5
- (B) 6
- (C) 12
- (D) 60
- (E) 1565

2. Consider the following instance variable and method

```
private int[] numList;

//Precondition: numList contains a list of int values in no particular order

public int mystery(int n)
{
    for (int k = 0; k <= numList.length - 1; k++)
    {
        if (n <= numList[k])
            return k;
    }
    return numList.length;
}
```

Which of the follow statements is most accurate about numList following the execution of the following statement?

```
int j = mystery(number);
```

- (A) The greatest value in numList is at index j.
- (B) The greatest value in numList that is less than number is at index j.
- (C) All values in numList from index 0 to j-1 are greater than or equal to number.
- (D) All values in numList from index 0 to j-1 are less than number.
- (E) All values in numList from index j to numList.length-1 are greater than number.

Questions 3–4 refer to the following incomplete class declaration for a new data type called a Quack.

```
Public class Quack
{

    // Constructor initializes myData
    public Quack()
    { /* implementation not shown */ }

    // Quack.
    public void enquack (Object x)
    { /* implementation not shown */ }

    // if front is true, returns the object at the front end of the Quack;
    // otherwise returns the object at the back end of the Quack. Assumes the Quack
    // is not empty.
    public Object dequack (boolean front)
    { /* implementation not shown */ }

    // Returns true if the Quack has no objects; otherwise returns false.
    public boolean isEmpty ()
    { /* implementation not shown */ }

    <designation> ArrayList myData;

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

3. Which of the following is the best choice for <designation> and the best reason for that choice?

- (A) <designation> should be private so that programs using a Quack will not be able to modify myData by using methods enquack and dequack, thereby preserving the principle of data stability.
- (B) <designation> should be private so that programs using a Quack can only modify myData by using methods such as enquack and dequack, thereby preserving the principle of information hiding.
- (C) <designation> should be private as an indication to programs using a Quack that myData can be modified directly but that it is *better* to modify myData only by using methods such as enquack and dequack, thereby preserving the principle of maximum information dissemination.
- (D) <designation> should be public because programs using a Quack need to know how the Quack class has been implemented in order to use it.
- (E) <designation> should be public. Otherwise, only objects constructed from derived subclasses of a Quack will be able to modify the contents of a Quack.

4. Which of the following is an effective return statement for isEmpty as described in the incomplete declaration above?

- (A) return (myData.length == 0)
- (B) return (size() == 0)
- (C) return (myData.size() == 0);
- (D) return (myData.length() == 0)
- (E) return (myData.size == 0)

5. Consider the following method definition:

```
public static int mystery(int n)
{
    if (n <= 1)
        return 2;
    else
        return 1 + mystery(n - 3);
}
```

Which of the following lines of code can replace the line in `mystery` containing the recursive call so that the functionality of `mystery` does not change?

- (A) `return 1 + ( (n + 2) / 3) ;`
- (B) `return 1 + ( (n + 3) / 2) ;`
- (C) `return 2 + ( (n + 1) / 3) ;`
- (D) `return 2 + ( (n + 2) / 3) ;`
- (E) `return 3 + ( (n + 1) / 2) ;`

Questions 6–7 refer to the following incomplete class declaration

```
public class DistanceTracker
{
    private int kilometers;
    private int meters;

    /** Constructs a DistanceTracker object
     * @param k the number of kilometers
     * Precondition:  $k \geq 0$ 
     * @param m the number of meters
     * Precondition:  $0 \leq m < 1000$ 
     */
    public DistanceTracker (int k, int m)
    {
        kilometer = k;
        meters = m;
    }
    /** @return the number of kilometers
     */
    public int getKilometers()
    { /* implementation not shown */ }
    /** @return the number of meters
     */
    public int getMeters()
    { /* implementation not shown */ }
    /** Adds k kilometers and m meters
     * @param k the number of kilometers
     * Precondition:  $k \geq 0$ 
     * @param m the number of meters
     * Precondition:  $m \geq 0$ 
     */
    public void addDistance(int k, int m)
    {
        kilometers += k;
        meters += m;
        /* rest of method not shown */
    }
    //Rest of class not shown
}
```

6. Which of the following code segments can be used to replace `/* rest of method not shown */` so the `addDistance` will correctly increase the distance?
- (A) `kilometers += meters / 1000`  
`meters = meters % 1000`
  - (B) `kilometers += meters % 1000`  
`meters = meters / 1000`
  - (C) `meters += kilometers % 1000`
  - (D) `kilometers += meters % 1000`
  - (E) `meters = meters % 1000`



7. Consider the following incomplete class declaration

```
public class DistanceTrackerSet
{
    DistanceTracker[] set;
    /*Declaration method not shown*/

    public DistanceTracker total()
    {
        DistanceTracker temp = new DistanceTracker(0, 0);
        for (int k = 0; k < set.length; k++)
        {
            /*missing code segment*/
        }
        return temp;
    }
    /*Other methods not shown*/
}
```

Assuming set is properly initialized with DistanceTracker objects and all needed classes are properly imported, which is the following can be used to replace `/*missing code segment*/` so that the method returns a DistanceTracker object with the total of all distances stored in set?

- (A) `temp.addDistance(temp[k].kilometers, temp[k].meters);`
- (B) `set[k].addDistance(temp[k].getKilometers(), temp[k].getMeters());`
- (C) `set[k].addDistance();`
- (D) `temp += temp.addDistance();`
- (E) `temp.addDistance(set[k].getKilometers(), set[k].getMeters());`

8. Consider the following method

```
public List<Integer> nums() {
    List<Integer> values = new ArrayList<Integer>() ;
    for (int i = 0; i < 50; i = i + 5)
        if (i % 4 == 1)
            values.add(i) ;
    return value ;
}
```

What will return of `nums()` contain?

- (A) [5, 45]
- (B) [5, 25, 45]
- (C) [0, 20, 40]
- (D) [5, 9, 13, 17, 21, 25, 29, 33, 37, 41, 45]
- (E) [0, 5, 10, 15, 20, 25, 30, 35, 40, 45]

9. Consider the following incomplete method `mystery`:

```
public static boolean mystery(boolean a, boolean b, boolean c)
{
    return <expression>;
}
```

What should `<expression>` be replaced with so that `mystery` returns `true` when exactly two of its three parameters are `true` and exactly one is `false` ; otherwise `mystery` returns `false`?

- (A) `(a && b && !c) | |`  
`(a && !b && c) | |`  
`(!a && b && c)`
- (B) `(a && b && !c) &&`  
`(a && !b && c) &&`  
`(!a && b && c)`
- (C) `(a | | b | | !c) &&`  
`(a | | !b | | c) &&`  
`(!a | | b | | c)`
- (D) `(a && b) | |`  
`(a && c) | |`  
`(b && c)`
- (E) `(a | | b) &&`  
`(a | | c) &&`  
`(b | | c)`

10. Consider the following code segment:

```
int x;
x = 5 - 4 + 9 * 12 / 3 - 10;
```

What is the value of `x` after the code segment is executed?

- (A) 13  
 (B) 27  
 (C) 30  
 (D) -57  
 (E) -10

11. What is the best way to declare a variable `myStrings` that will store 50 `String` values if each `String` will be no longer than 25 characters?

- (A) `ArrayList <String> myStrings[String[50]] ;`  
 (B) `ArrayList <String> myStrings = new String[50] ;`  
 (C) `ArrayList <String> myStrings = new String[25] ;`  
 (D) `String [] myStrings = new String [50, 25] ;`  
 (E) `String [] myStrings = new String [50] ;`

12. Consider the following code segment

```
List <Integer> scores= new ArrayList<Integer>() ;
scores.add(93) ;
scores.add(97) ;
scores.add(84) ;
scores.add(91) ;
scores.remove(2) ;
scores.add(1, 83) ;
scores.set(3, 99) ;
System.out.println(scores) ;
```

What is the output of the code segment?

- (A) [83, 93, 99, 91]
  - (B) [93, 83, 91, 99]
  - (C) [83, 94, 91, 99]
  - (D) [93, 83, 97, 99]
  - (E) The code throws an `ArrayIndexOutOfBoundsException`
13. Consider the following precondition, postcondition, and signature for the `getDigit` method:

```
// precondition: n >= 0
//           whichDigit >= 0
// postcondition: Returns the digit of n in
//           the whichDigit position
//           when the digits of n are
//           numbered from right to
//           left starting with zero.
//           Returns 0 if whichDigit >=
//           number of digits of n.
int getDigit (int n, int whichDigit)
```

Consider also the following three possible implementations of the `getDigit` method:

- I. 

```
if (whichDigit == 0)
    return n % 10;
else
    return getDigit (n / 10, whichDigit - 1) ;
```
- II. 

```
return (n / (int) Math.pow(10, whichDigit) ) % 10;
```
- III. 

```
for (int k = 0; k < whichDigit; k++)
    n /= 10;
return n % 10;
```

Which implementation(s) would satisfy the postcondition of the `getDigit` method?

- (A) I and II
- (B) I and III
- (C) II and III
- (D) I, II, and III
- (E) None of the above

14. Consider an array of integers.

4      10      1      2      6      7      3      5

Assume that `SelectionSort` is used to order the array from smallest to largest values.

Which of the following represents the state of the array immediately after the first iteration of the outer for loop in the `SelectionSort` process?

- (A) 1      4      10      2      3      6      7      5  
 (B) 1      2      4      6      10      7      3      5  
 (C) 1      10      4      2      6      7      3      5  
 (D) 4      3      1      5      6      7      10      2  
 (E) 5      3      7      6      2      1      10      4

15. Assume that a program declares and initializes `v` as follows

```
String [] v;  
v = initialize () ;           // Returns an array of  
                               // length 10 containing  
                               // ten valid strings
```

Which of the following code segments correctly traverses the array *backwards* and prints out the elements (one per line)?

```
I. for (int k = 9; k >= 0; k --)  
    System.out.println(v[k]) ;  
II. int k = 0;  
    while (k < 10)  
    {  
        System.out.println(v[9-k]);  
        k++;  
    }  
III. int k = 10;  
    while (k >= 0)  
    {  
        System.out.println(v[k]);  
        k -- ;  
    }
```

- (A) I only  
 (B) II only  
 (C) I and II only  
 (D) II and III only  
 (E) I, II, and III

16. Consider the following method

```
/**Precondition: @param set  an ArrayList that contains distinct integers
 *@param n an int value
 */
public int mystery(List<Integer> set, int n)
{
    for (int k = 0; k < set.size(); k++)
    {
        if (set.get(k) > n)
        {
            return (set.remove(k) + mystery(set, n));
        }
    }
    return 0;
}
```

What is returned by the method call `mystery(set, n)`?

- (A) 0
- (B) The number of elements of set that are greater than n
- (C) The sum of the elements of set that are greater than n
- (D) The sum of the elements of set that are less than n
- (E) The sum of the elements of set that are less than or equal to n

17. Consider the following two-dimensional array

```
[ [0, 0, 0 ,0]
  [0, 1, 0, 0]
  [0, 1, 2, 0]
  [0, 1, 2, 3] ]
```

Which of the following methods returns this two-dimensional array?

- (A) 

```
public int[][] nums()
{
    int[][] temp = new int[4][4] ;
    for (int j = 0; j < 4; j++)
    {
        for (int k = 0; k < 4; k++)
        {
            temp[j][k] = j ;
        }
    }
    return temp ;
}
```
- (B) 

```
public int[][] nums()
{
    int[][] temp = new int[4][4] ;
    for (int j = 0; j < 4; j++)
    {
        for (int k = 0; k < 4; k++)
        {
            temp[j][k] = k ;
        }
    }
    return temp ;
}
```
- (C) 

```
public int[][] nums()
{
    int[][] temp = new int[4][4] ;
    for (int j = 0; j < 4; j++)
    {
        for (int k = j; k < 4; k++)
        {
            temp[j][k] = k ;
        }
    }
    return temp ;
}
```

```
(D) public int[][] nums()
{
    int[][] temp = new int[4][4] ;
    for (int j = 0; j < 4; j++)
    {
        for (int k = 0; k <= j; k++)
        {
            temp[j][k] = j ;
        }
    }
    return temp ;
}

(E) public int[][] nums()
{
    int[][] temp = new int[4][4] ;
    for (int j = 0; j < 4; j++)
    {
        for (int k = 0; k <= j; k++)
        {
            temp[j][k] = k ;
        }
    }
    return temp ;
}
```

18. A children's club classifies members based on age according to the table below

Years	Classification
Under 3	Infant
3 to 7 inclusive	Pee-wee
8 to 13 inclusive	Cub
Over 14	Leader

Which of the following methods will correctly take the integer parameter age and return the String Classification?

- (A) 

```
public String Classification(int age)
{
    String temp ;
    if (age < 3)
        temp = "Infant" ;
    if (age <= 7)
        temp = "Pee-Wee" ;
    if (age <= 13)
        temp = "Cub" ;
    if (age >= 14)
        temp = "Leader" ;
    return temp;
}
```
- (B) 

```
public String Classification(int age)
{
    String temp ;
    if (age < 3)
        temp = "Infant" ;
    if (3 <= age <= 7)
        temp = "Pee-Wee" ;
    if (8 <= age <= 13)
        temp = "Cub" ;
    if (age >= 14)
        temp = "Leader" ;
    return temp;
}
```
- (C) 

```
public String Classification(int age)
{
    String temp ;
    if (age < 3)
        temp = "Infant" ;
    else if (age <= 7)
        temp = "Pee-Wee" ;
    else if (age <= 13)
        temp = "Cub" ;
    else if (age > 14)
        temp = "Leader" ;
    return temp;
}
```



```
(D) public String Classification(int age)
{
    String temp ;
    if (age < 3)
        temp = "Infant" ;
    else if (age < 7)
        temp = "Pee-Wee" ;
    else if (age < 13)
        temp = "Cub" ;
    else if (age > 14)
        temp = "Leader" ;
    return temp;
}

(E) public String Classification(int age)
{
    String temp ;
    if (age < 3)
        temp = "Infant" ;
    if (age < 7)
        temp = "Pee-Wee" ;
    if (age < 13)
        temp = "Cub" ;
    if (age > 14)
        temp = "Leader" ;
    return temp;
}
```

For Questions 19–20 Consider the method `getGap` with line numbers added for reference. Method `getGap` is intended to find the maximum difference between the indexes of any two occurrence of `num` in the array `arr`. The method `getGap` does not work as intended.

For example, if the array `arr` contains `[8, 7, 5, 5, 4, 3, 2, 7, 1, 2, 7]`, the call `getGap(arr, 7)` should return 9, the difference between the indexes of the first and last occurrence of 7.

/\*\*Precondition: `arr` contains at least two occurrences of `num` \*/

Line 1:	<code>public int getGap(int[] arr, int num)</code>
Line 2:	<code>{</code>
Line 3:	<code>    int index1 = -1;</code>
Line 4:	<code>    int index2 = -1;</code>
Line 5:	<code>    for (int k = 0; k &lt; arr.length; k++)</code>
Line 6:	<code>    {</code>
Line 7:	<code>        if (arr[k] == num)</code>
Line 8:	<code>        {</code>
Line 9:	<code>            if (index1 == -1)</code>
Line 10:	<code>            {</code>
Line 11:	<code>                index1 = k;</code>
Line 12:	<code>                index2 = k;</code>
Line 13:	<code>            }</code>
Line 14:	<code>        else</code>
Line 15:	<code>        {</code>
Line 16:	<code>            index1 = index2;</code>
Line 17:	<code>            index2 = k;</code>
Line 18:	<code>        }</code>
Line 19:	<code>    }</code>
Line 20:	<code>}</code>
Line 21:	<code>    return (index2 - index1);</code>
Line 22:	<code>}</code>

19. The method `getGap` does not work as intended. Which of the following best describes the return of the method `getGap` ?

- (A) The difference between the indexes of the last two occurrences of `num` in `arr`.
- (B) The minimum difference between the indexes of any two occurrences of `num` in `arr`.
- (C) The difference between the first two occurrences of `num` in `arr`
- (D) The length of the array `arr`
- (E) The number of occurrences of `num` in `arr`

20. Which of the following changes should be made to `getGap` so that the method will work as intended?

- (A) Delete the statement at line 4.
- (B) Delete the statement at line 11.
- (C) Delete the statement at line 12.
- (D) Delete the statement at line 16.
- (E) Delete the statement at line 17.

Questions 21–23 refer to the following incomplete class declaration used to represent calendar dates.

```
Public class Date
{
    private int month;
        // represents month 0-11
    private int day;
        // represents day of the month
        // 0-31
    private int year;
        // represents the year

    // constructor sets the private data
    public Date (int m, int d, int y)
    { /* implementation not shown */ }

    // postconditions: returns the month
    public int getMonth()
    { /* implementation not shown */ }

    // postcondition: return the day
    public int getDay()
    { /* implementation not shown */ }

    // postcondition: returns the year
    public int getYear()
    { /* implementation not shown */ }

    // postcondition: returns the number of
    //                 days which, when
    //                 added to this Date
    //                 gives newDate
    public int daysUntil (Date newDate)
    { /* implementation not shown */ }

    // postcondition: returns true if
    //                 the month, day, and
    //                 year of this Date are
    //                 are equal to those of
    //                 other; otherwise
    //                 returns false
    public boolean equals (Date other)
    { /* implementation not shown */ }

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

21. Consider the method equals of the Object class.

Which of the following method signatures is appropriate for the equals method?

- (A) public boolean equals (Object other)
- (B) public int equals (Object other)
- (C) public boolean equals (Date other)
- (D) public int equals (Date other)
- (E) public boolean equals (Date d1, Date d2)

**GO ON TO THE NEXT PAGE.**

22. Which of the following code segments could be used to implement the equals method of the Date class so that the equals method works as intended?

- I.   if (month == other.month)  
      if (day == other.day)  
          if (year == other.year)  
              return true;  
      Return false;
- II.   if (month == other.getMonth() &&  
      day == other.getDay() &&  
      year == other.getYear())  
      return true;  
      else  
      return false;
- III. return ! ((getMonth() != other.getMonth()) | |  
      (getDay() != other.getDay()) | |  
      (getYear() != other.getYear())) ;

- (A) I only  
(B) II only  
(C) I and II only  
(D) II and III  
(E) I, II, and III
23. During the testing of the Date class, it is determined that the class does not correctly handle leap years—although it handles non-leap years correctly.

In which method of the Date class is the problem most likely to be found?

- (A) the Date constructor  
(B) the getMonth method  
(C) the getDay method  
(D) the daysUntil method  
(E) the equals method

24. Consider the following methods:

```
public static void mystery ()
{
    int [] A;
    A = initialize ();
    // returns a valid initialized
    // array of integers
    for (int k = 0; k < A.length / 2; k++)
        swap (A[k], A[A.length - k - 1]);
}

public static void swap (int x, int y)
{
    int temp;
    temp = x;
    x = y;
    y = temp;
}
```

Which of the following best characterizes the effect of the for loop in the method mystery?

- (A) It sorts the elements of A.
- (B) It reverses the elements of A.
- (C) It reverses the order of the first half of A and leaves the second half unchanged.
- (D) It reverses the order of the second half of A and leaves the first half unchanged.
- (E) It leaves all of the elements of A in their original order.

25. Consider the following code segment:

```
int [][] A = new int [4][3] ;
for (int j = 0; j < A[0].length; j++)
    for (int k = 0; k < A.length; k++)
        if (j == 0)
            A[k][↓] = 0;
        else if (k % j == 0)
            A[k][j] = 1;
        else
            A[k][j] = 2;
```

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

- (A) 0 0 0 0  
1 1 1 1  
1 2 1 2
- (B) 0 1 1 1  
0 2 2 2  
0 1 2 1
- (C) 0 0 0 0  
1 1 2  
1 1 1  
1 1 2
- (D) 0 1 1  
0 2 1  
0 2 2  
0 2 1
- (E) 0 1 1  
0 1 2  
0 1 1  
0 1 2

26. Consider the following method:

```
/** @param num an int value such that num >= 0
 */
public void mystery(int num)
{
    System.out.print(num % 100);
    if ((num / 100) != 0)
    {
        mystery(num / 100);
    }
    System.out.print(num % 100);
}
```

Which of the following is printed as a result of the call `mystery(456789)`?

- (A) 456789
- (B) 896745
- (C) 987654
- (D) 456789896745
- (E) 896745456789

27. Consider the following method:

```
public static int mystery (int x, int y)
{
    if (x > 0)
        return x;
    else if (y > 0)
        return y;
    else
        return x / y;
}
```

In accordance with good design and testing practices, which of the following is the best set of test cases ( $x$ ,  $y$ ) for the method `mystery`?

- (A) (3, 4), (-3, 4), (-3, -4)
- (B) (3, 4), (-3, 4), (-3, -4), (-3, 0)
- (C) (3, 4), (3, -4), (-3, -4), (-3, 0)
- (D) (3, 4), (3, -4), (-3, -4), (-3, 4), (-3, 0)
- (E) (3, 4), (2, 5), (3, -4), (-3, 0), (4, 0), (0, 0)

28. Consider the following method.

```
/** Precondition: numList is not empty
 */
private int mystery(int[] numList)
{
    int n = numList.length - 1;
    for (int k : numList)
    {
        if (numList[n] > numList[k])
        {
            n = k;
        }
    }
    return numList[n];
}
```

Which of the following best describes the return of `mystery`?

- (A) The largest value in the array `numList`
- (B) The least value in the array `numList`
- (C) The index of the largest value in the array `numList`
- (D) The index of the least value in the array `numList`
- (E) The number of indexes whose values are less than `numList[n]`

29. Consider the following method.

```
public int[] editArray(int[] arr, int old, int new)
{
    /*missing code*/
    return arr;
}
```

The method above is intended to replace any instance of old in arr with any instances of new. Which of the following can be used to replace `/*missing code*/` to replace any values of old in the array with values of new?

- (A) 

```
for (int k = 0; k < arr.length; k++)
{
    if (arr[k] = old)
    {
        arr[k] == new;
    }
}
```
- (B) 

```
for (int k = 0; k < arr.length; k++)
{
    if (arr[k] == old)
    {
        arr[k] = new;
    }
}
```
- (C) 

```
while (arr[k] == old)
{
    arr[k] = new
}
```
- (D) 

```
for (int k = 0; k < arr.length; k++)
{
    arr[k] == new;
}
```
- (E) 

```
while (int k = 0; k < arr.length; k++)
{
    if (arr[k] = old)
    {
        arr[k] == new;
    }
}
```



30. Consider the following two classes.

```
public class SalesPerson
{
    public void sale()
    {
        System.out.print("greet ");
        pitch();
    }
    public void pitch()
    {
        System.out.print("pitch ");
    }
}

public class CommissionedSalesPerson extends SalesPerson
{
    public void sale()
    {
        super.sale();
        System.out.print("record ");
    }
    public void pitch()
    {
        super.pitch();
        system.out.print("close ");
    }
}
```

The following code segment is found in a class other than SalesPerson.

```
SalesPerson vincent = new CommissionedSalesPerson();
vincent.sale();
```

Which of the following is the best description of the functionality of this code segment?

- (A) greet pitch
- (B) greet pitch close
- (C) greet pitch record
- (D) greet pitch record close
- (E) greet pitch close record

31. Consider the following declaration of a class that will be used to represent dimensions of rectangular crates.

```
public class Crate
{
    private int length;
    private int width;
    private int height;

    public Crate(int x, int y, int z)
    {
        length = x;
        width = y;
        height = z;
    }

    //other methods not shown
}
```

The following incomplete class declaration is intended to extend the `Crate` class so that the color of the crate can be specified.

```
public class ColoredCrate{
    private String color;
    //Constructors not shown
    //Other methods not shown
}
```

Which of the following possible constructors for `ColoredCrate` would be considered legal?

- I. 

```
public ColoredCrate(int a, int b, int c, String crateColor)
{
    length = a;
    width = b;
    height = c;
    color = crateColor;
}
```
- II. 

```
public ColoredCrate(int a, int b, int c, String crateColor)
{
    super (a, b, c)
    color = crateColor;
}
```
- III. 

```
public ColoredCrate()
{
    color = "";
}
```

- (A) I only
- (B) III only
- (C) I and II only
- (D) I and II only
- (E) II and III only

32. Consider the following three proposed implementations of method reverse, intended to return the reverse the order of objects in an ArrayList:

- I. 

```
public static ArrayList<Object> reverse (ArrayList<Object> q)
{
    ArrayList<Object> s = new ArrayList<Object>();
    while (q.size() != 0)
        s.add(0, q.remove(0));
    return s;
}
```
- II. 

```
public static ArrayList<Object> reverse (ArrayList<Object> q)
{
    ArrayList<Object> s = new ArrayList<Object>(s);
    for (int k = 0; k < q.size(); k++)
        s.add(0, q.remove(0));
    return s;
}
```
- III. 

```
public static ArrayList<Object> reverse (ArrayList<Object> q)
{
    Object obj;
    if (q.size() != 0)
    {
        obj = q.remove(0);
        q = reverse(q);
        q.add(obj);
    }
    return q;
}
```

Which of the above implementations of method reverse work as intended?

- (A) I only  
 (B) III only  
 (C) I and II  
 (D) I and III  
 (E) I, II, and III

33. Consider the following code segment.

```
List<Integer> values = new ArrayList<Integer>() ;
values.add(5) ;
values.add(3) ;
values.add(2) ;
values.add(2) ;
values.add(6) ;
values.add(3) ;
values.add(9) ;
values.add(2) ;
values.add(1) ;
for (int j = 0; j < values.size(); j++)
{
    if (values.get(j).intValue() == 2)
    {
        values.remove(j);
    }
}
```

What will values contain as a result of executing this code segment?

- (A) [5, 3, 2, 2, 6, 3, 9, 2, 1]
- (B) [5, 3, 2, 6, 3, 9, 1]
- (C) [5, 3, 6, 3, 9, 1]
- (D) [2, 2, 2, 5, 3, 6, 3, 9, 1]
- (E) The code throws an `ArrayIndexOutOfBoundsException` exception

34. Consider the class `Data` partially defined below. The completed `max1D` method returns the maximum value of `b`, a one-dimensional array of integers. The completed `max2D` method is intended to return the maximum value `c`, a two dimensional array of integers.

```
public class Data
{
    /** Returns the maximum value of one-dimensional array b */
    public int max1D(int[] b)
    { /* implementation not shown */}
    /** Returns the maximum value of two-dimensional array c */
    public int max2D(int[] c)
    {
        int max;
        /* missing code */
        returns max
    }
    /* other methods of Data class not shown*/
}
```

Assume that `max1D` works as intended. Which of the follow can replace `/* missing code */` so that `max2D` works as intended.

- I. 

```
for (int[] row: c)
{
    max = max1D(row);
}
```
- II. 

```
max = max1D(c[0]);
for (int k = 1; k <= c.length; k++)
{
    max = max1D(c[k]);
}
```
- III. 

```
max = max1D(c[0]);
for (int[] row: c)
{
    if (max < max1D(row))
    {
        max = max1D(row);
    }
}
```

- (A) I only
- (B) III only
- (C) I and II only
- (D) II and III only
- (E) I, II, and III only

35. Consider the following instance variable, numList, and incomplete method, countZeros. The method is intended to return an integer array count such that for all k, count[k] is equal to the number of elements from numList[0] through numList[k]. For example, if numList contains the values {1, 4, 0, 5, 0, 0}, the array countZeros contains the values {0, 0, 1, 1, 2, 3}.

```
public int[] countZeros(int[] numList)
{
    int[] count = new int[numList.length];
    for (int k : count)
    {
        count[k] = 0;
    }
    /* missing code */
    return count;
}
```

The following two versions of /\*missing code\*/ are suggested to make the method work as intended.

Version 1

```
for (int k = 0; k <= numList.length; k++)
{
    for (int j = 0; j <= k; j++)
    {
        if (numList[j] == 0)
        {
            count[k] = numList[k] + 1;
        }
    }
}
```

Version 2

```
for (int k = 0; k < numList.length; k++)
{
    if (numList[k] == 0)
    {
        count[k] = count[k - 1] + 1;
    }
    else
    {
        count[k] = count[k - 1];
    }
}
```

Which of the following statements are true?

- (A) Both Version 1 and Version 2 will work as intended, but Version 1 is faster than Version 2.
- (B) Both Version 1 and Version 2 will work as intended, but Version 1 is faster than Version 2.
- (C) Version 1 will work as intended but Version 2 causes an `ArrayIndexOutOfBoundsException`.
- (D) Version 2 will work as intended but Version 1 causes an `ArrayIndexOutOfBoundsException`.
- (E) Version 1 and Version 2 each cause an `ArrayIndexOutOfBoundsException`.

36. A real estate agent wants to develop a program to record information about apartments for rent. For each apartment, she intends to record the number of bedrooms, number of bathrooms, whether pets are allowed, and the monthly rent charged. Which of the following object oriented program designs would be preferred?
- (A) Use a class Apartment with four subclasses: Bedrooms, Bathrooms, PetsAllowed, and Rent.
  - (B) Use four classes: Bedrooms, Bathrooms, PetsAllowed, and Rent, each with subclass Apartment.
  - (C) Use of class Apartment with four instance variables `int bedrooms`, `int bathrooms`, `boolean petsAllowed`, and `double rent`.
  - (D) Use five unrelated classes: Apartment, Bedrooms, Bathrooms, PetsAllowed, and Rent
  - (E) Use a class Apartment, with a subclass Bedrooms, with a subclass Bathrooms, with a subclass PetsAllowed, with a subclass Rent.

37. Consider the following declarations

```
public class Book
{
    boolean hasMorePagesThan(Book b);
    //other methods not shown
}
Public class Dictionary extends Book
{
    //other methods not shown
}
```

Of the following method headings of `hasMorePagesThan`, which can be added to Dictionary so that it will satisfy the Book superclass?

- I. `int hasMorePagesThan(Book b)`
- II. `boolean hasMorePagesThan(Book b)`
- III. `boolean hasMorePagesThan(Dictionary d)`

- (A) I only
- (B) I and II only
- (C) II only
- (D) II and III only
- (E) I, II, and III

38. Consider the following method.

```
/**Precondition: set contains does not contain any negative values
 */
public int mystery(int[] set, int max)
{
    int m = 0;
    int count = 0;
    for (int n = 0; n < set.length && set[n] < max; n++)
    {
        if (set[n] >= m)
        {
            m = set[n]; //Statement A
        }
        count++; //Statement B
    }
    return count;
}
```

Assume that mystery is called and is executed without error. Which of the following are possible combinations of the number of the value of max, the number of times Statement A is executed and the number of times Statement B is executed?

	Value of max	Executions of Statement A	Executions of Statement B
I	8	2	3
II	3	7	5
III	7	0	4

- (A) I only
- (B) III only
- (C) I and II only
- (D) I and III only
- (E) I, II, and III



39. The following method is intended to return an array that inserts an integer  $m$  at index  $n$ , pushing the values of all indexes after  $n$  to the index one higher. For example, if the index `arr` is

4	2	1	3
---	---	---	---

and the command `arr = insert(arr, 5, 2)` is called, the method is intended to return

4	2	5	1	3
---	---	---	---	---

Line 1	<code>public int[] insert(int[] arr, int m, int n)</code>
Line 2	<code>{</code>
Line 3	<code>int[] temp = new int[arr.length+1];</code>
Line 4	<code>for (int k = 0; k &lt; arr.length; k++)</code>
Line 5	<code>{</code>
Line 6	<code>if (k &lt; n)</code>
Line 7	<code>{</code>
Line 8	<code>temp[k] = arr[k];</code>
Line 9	<code>}</code>
Line 10	<code>else</code>
Line 11	<code>{</code>
Line 12	<code>temp[k + 1] = arr[k];</code>
Line 13	<code>}</code>
Line 14	<code>}</code>
Line 15	<code>temp[m] = n;</code>
Line 16	<code>return temp;</code>
Line 17	<code>}</code>

The method `insert` does not work as intended. Which of the following changes will cause it to work as intended?

- (A) Change Line 6 to `if (k > n)`
- (B) Change Line 6 to `if (k <= n)`
- (C) Change Line 12 to `temp[k] = arr[k + 1];`
- (D) Change Line 15 to `temp[n] = m;`
- (E) Change Line 16 to `return arr;`

40. If X, Y, and Z are integer values, the boolean expression

$$(X > Y) \ \&\& \ (Y > Z)$$

can be replaced by which of the following?

- (A)  $X > Z$
- (B)  $(X < Y) \ || \ (Y < Z)$
- (C)  $(X \leq Y) \ || \ (Y \leq Z)$
- (D)  $! \ ( (X < Y) \ || \ (Y < Z) )$
- (E)  $! \ ( (X \leq Y) \ || \ (Y \leq Z) )$

**END OF SECTION I**

**IF YOU FINISH BEFORE TIME IS CALLED,  
YOU MAY CHECK YOUR WORK ON THIS SECTION.**

**DO NOT GO ON TO SECTION II UNTIL YOU ARE TOLD TO DO SO.**