

ICS45J Sample Exam Questions

To help you study for the midterm and final, here are some questions from previous exams I gave in Java programming courses I've taught. (Since this is the first time this course is being offered, I have no "old" ICS 45J exams questions to share.) These questions are not necessarily all-inclusive of the subject matter. There may be questions on the actual exams asking about material that no question here asks about. Likewise, there will be questions here that do not pertain to the particular test for which you may be studying--for example, if you are getting ready for the midterm, you will find questions here about material that will be covered by the final; just ignore those questions for the present but, of course, don't ignore them when studying for the final!

For all program fragments, assume any necessary "imports" are present, and that the fragments are a part of an otherwise perfect program.

1. Which of these statements about constructors is *false*?
 - A. A constructor has no return type
 - B. Its name must be the same as the class in which it is defined
 - C. There must be exactly one constructor defined for a class
 - D. Constructors are almost always declared as public
 - E. They can appear anywhere in the class where it is legal to declare a method
2. When does Java know an object is no longer needed? And what happens to an unneeded object's storage?
 - A. The programmer tells Java when an object is no longer needed by calling `dispose()` on it; the object's memory is released back to the memory pool.
 - B. If there are no references to the object, Java knows the object is no longer needed and automatically returns its memory to the memory pool.
 - C. If there are no references to the object, Java marks it as no longer needed; the memory stays in use until the programmer explicitly returns it to the memory pool.
 - D. Objects, once constructed, stay active until the program terminates, so though the programmer may know an object is no longer needed, Java does not know this; objects' memory is returned to the memory pool when the program terminates.
 - E. Objects, once constructed, stay active until the method in which they were constructed terminates, so though the programmer may know an object is no longer needed, Java does not know this; objects' memory is returned to the memory pool when the method terminates.

3. Which of the following Java statements set `even` to true if `n` is even, and to false if `n` is odd? (`n` is an integer.) Assume `n >= 0`. (Even numbers are those integers which, when divided by 2, have a remainder of 0.)

- I. `boolean even = (n/2.0 == (double)(n/2));`
- II. `boolean even = (n % 2 == 0);`
- III. `boolean even = (n div 2 == 0);`
- IV. `boolean even = (n % 2 == n/2);`

- A. I only
- B. I and II only
- C. II and III only
- D. III and IV only
- E. I, II and IV only

4. Which of the following expressions is equivalent to the boolean expression

`!(A < 5 && B != C)`

- A. `A > 5 || B != C`
- B. `A >= 5 && B == C`
- C. `!(A < 5) || (B != C)`
- D. `A >= 5 || B == C`
- E. `A < 5 && B == C`

5. What is the output of this program fragment? Read it carefully!

```
String greet = "Hi";
String name = "Smedley";
String nickName = name.substring(0,4);
if (nickName == name.substring(0,4));
    System.out.println("has real nickname");
else if (greet + name == greet + nickName)
    System.out.println("no real nickname");
else
    System.out.println("hmmm...changed names?");
```

- A. has real nickname
- B. no real nickname
- C. hmmm...changed names?
- D. it's one of the three lines given in A, B, and C above, we can't tell which one without running the program
- E. none, because there is at least one compile-time error

When answering the next 5 questions, consider this code fragment:

```
int sum = 0;
int i = 0;
while (i < 5)
{
    sum = sum + i;
    i++;
}
System.out.print(i);
System.out.print(" ");
System.out.print(sum);
```

6. What is the value of `i` when `System.out.print(i)` is executed?
 - A. 6
 - B. 5
 - C. 4
 - D. 3
 - E. unpredictable, since `i` is local to the loop

7. What is the value of `sum` when `System.out.print(sum)` is executed?
 - A. 6
 - B. 10
 - C. 15
 - D. 21
 - E. unpredictable, since `sum` is dependent upon a variable local to the loop

8. The fragment executes more times through the loop than is necessary. What change to the fragment can be made that removes the unneeded passes through the loop and doesn't change the values of `i` and `sum` that are printed out?
 - A. initialize `sum` to 1, rather than 0
 - B. initialize `i` to 1, rather than 0
 - C. make the while loop boolean (`i <= 4`), instead of (`i < 5`)
 - D. make the while loop boolean (`i < 4`), instead of (`i < 5`)
 - E. place the statement `i++;` before `sum = sum + i;` (rather than after it)

9. Suppose we replace the while loop in the fragment above with a do loop. Which of the following do loops will result in the same value of Sum printing out?

- A. do
 {
 i++;
 sum = sum + i;
 }
 while (i <= 5);
- B. do
 {
 i++;
 sum = sum + i;
 }
 while (i < 5);
- C. do
 {
 sum = sum + i;
 i++;
 }
 while (i < 5);
- D. do
 {
 sum = sum + i;
 i++;
 }
 while (i <= 5);
- E. Both A and C

10. Suppose we replace the while loop in the fragment above with a for loop. Which of the following for loops will result in the same value of Sum printing out?

- A. for (int i = 0; i <= 5; i++)
 sum = sum + i;
- B. for (int i = 1; i <= 5; i++)
 sum = sum + i;
- C. for (int i = 1; i < 5; i++)
 sum = sum + i;
- D. for (int i = 2; i < 5; i++)
 sum = sum + i;
- E. for (int i = 1; i < 6; i++)
 sum = sum + i;

11. According to Jacobson, when is it good practice to use a break statement?

- A. As appropriate to break out of a switch statement
B. As needed to break out of a loop
C. As needed to break out of a deeply nested series of if statements
D. Both A and B.
E. Both A and C.

12. What is the output from the following Java program fragment? Read it carefully!

```
String S = ""; String T = "";
int i = 4;
for (i = 1; i <= 3; i++);
    S = S + "!";
for (i = 1; i < 4; i++)
    T = T + "*";
System.out.print(S);
System.out.println(T);
```

- A. (that is, the empty string, printed twice)
- B. *
- C. !***
- D. !****
- E. !!!***

13. Which `if` statement below is equivalent to the given `switch` statement (that is, produces the same output under the same conditions)? Assume `answer` is a previously declared `int`.

```
switch (answer)
{ case 0: System.out.print("0 entered"); break;
  case 1: System.out.print("1 entered"); break;
  case 3: System.out.print("3 entered"); break;
  case 5: System.out.print("5 entered"); break;
  default: System.out.print("Other value entered");
}
```

- I.

```
if (answer == 0 || 1 || 3 || 5)
    System.out.print(answer + " entered");
else
    System.out.print("Other value entered");
```
- II.

```
if (answer = 0 || answer = 1 || answer = 3 || answer = 5)
    System.out.print(answer + " entered");
else System.out.print("Other value entered");
```
- III.

```
if ((answer >= 1 && answer <= 5 && answer % 2 == 1) ||
    (answer == 0))
    System.out.print(answer + " entered");
else System.out.print("Other value entered");
```
- IV.

```
if (answer == 0)
    System.out.print("0 entered");
if (answer == 1)
    System.out.print("1 entered");
if (answer == 3)
    System.out.print("3 entered");
if (answer == 5)
    System.out.print("5 entered");
if (answer != 0 && answer != 1 &&
    answer != 3 && answer != 5)
    System.out.print("Other value entered");
```

```
V. if (answer == 0)
    System.out.print("0 entered");
else if (answer == 1)
    System.out.print("1 entered");
else if (answer == 3)
    System.out.print("3 entered");
else if (answer == 5)
    System.out.print("5 entered");
else if (answer != 0 & 1 & 3 & 5)
    System.out.print("Other value entered");
```

- | | |
|--------------------|-------------------|
| A. I and III only | D. II and V only |
| B. III and IV only | E. II and IV only |
| C. IV and V only | |

When answering the next 5 questions, consider this program; comments indicate where missing needed components of the program are to be placed.

```
public class MainClass
{
    //definition of a function that prints out a greeting

    public static void main(String[] args)
    {
        //(2) print the greeting
        //(3) construct a MyClass object called myObject
        //(4) update myObject
        //    print myObject
    }
}

class MyClass
{
    // (1) definition of MyClass constructor

    public static void greetings()
    {
        // definition of greets
    }

    public void update(int num, String title)
    {
        // definition of update
    }

    public void print()
    {
        // definition of print
    }

    private int numOfItems;
    private String reportTitle;
}
```

14. Suppose you are writing the definition of `MyClass` (line (1) above). Which of the following function signatures (headers) is correct?

- A. `public MyClass`
- B. `public MyClass()`
- C. `public void MyClass`
- D. `public void MyClass()`
- E. `public MyClass(void)`

15. Suppose you wish to call the method that prints the greeting, at line(2) above. Which of the following statements will call this method correctly? `myObject` is the `MyClass` object defined in the question above.

- A. `MainClass.greetings();`
- B. `myObject.greetings();`
- C. `MyClass.greetings();`
- D. `void result = greetings();`
- E. `greetings();`

16. Suppose you wish to construct a `MyClass` object called `myObject` at line (3) above. Which of the following statements will correctly do this?

- A. `MyClass myObject;`
- B. `myObject.MyClass();`
- C. `MyClass myObject = MyClass();`
- D. `MyClass myObject = new(MyClass);`
- E. `MyClass myObject = new MyClass();`

17. Suppose you wish to call the `update` method at line(4) above. Which of the following statements will call this method correctly?

- A. `update(myObject(3, "Hi!"));`
- B. `update(3, "Hi!");`
- C. `MyClass.myObject.update(3, "Hi!");`
- D. `myObject.update(3, "Hi!");`
- E. `MyClass.update(3, "Hi!");`

18. Here's `update`, completed. The intent of the method is to update the class' fields with the values provided by the parameters `num` and `title`:

```
public void update(int num, String title)
{
    int numOfItems = num;
    reportTitle = title;
}
```

If this method is called, what can we say about the values of the member variables `numOfItems` and `reportTitle` when that call completes?

- A. `numOfItems` takes on the value of `num`; `reportTitle` takes on the value of `title`.
- B. `numOfItems` takes on the value of `num`; `reportTitle`'s value remains unchanged.
- C. `numOfItems` value remains unchanged; `reportTitle` takes on the value of `title`.
- D. Both `numOfItems`' and `reportTitle`'s values remain unchanged.
- E. A run-time error occurs because of the illegal redefinition of a variable inside the function.

Use this method when answering the next 2 questions.

```
public static int theValue()
{
    boolean notDone = true;
    int foundIt = -9;
    do {
        System.out.print("Enter score: ");
        int cntScore = getInt(); //assume getInt() is defined as in lab
        if (cntScore != -99)
        {   if ((cntScore >= 0) && (cntScore <= 100))
            if (cntScore > foundIt)
                foundIt = cntScore;
        }
        else
            notDone = false;
    } while (notDone == true);
    return foundIt;
}
```

19. Which aspects of the fragment (of those listed below) violate good style standards (as defined in this course)?

- I. having the { after do (on the same line)—the { should be on its own line directly underneath do
- II. The constants -9, -99, 0 and 100 should have been named, and the names should have been used in the fragment
- III. Using notDone == true as the test for the loop—there is a cleaner expression that can be used
- IV. The user is not told the sentinel value

- A. I and II only
- B. III and IV only
- C. I, II and III only
- D. II, III and IV only
- E. I through IV

20. Which statement best describes the intended purpose of this fragment?

- A. to compute the minimum of a group of user-entered (integer) scores
- B. to compute the maximum of a group of user-entered (integer) scores
- C. to inform the user when s/he enters a score that is not between 0 and 100
- D. to compute the maximum of the entered scores that are between 0 and 100
- E. to compute the minimum of the entered scores that are between 0 and 100

21. Which of the following statements about Java arrays and ArrayLists are true?

- I. Arrays are similar to objects, but technically are not true objects.
 - II. Once an ArrayList's size is set, it cannot be changed without reconstructing it.
 - III. Arrays can directly hold primitive types as well as object references.
 - IV. Array indexing begins at 0, but ArrayList indexing begins at 1.
- A. I and III only
 - B. I, II and III only
 - C. II and III only
 - D. I, II and IV only
 - E. I, II, III and IV

22. Consider a program written using an array A of size 50. Now suppose you want to change the program so that A is an ArrayList. What changes must or should be made to the program? Your goal is to have a correctly functioning program that follows good programming practice, while minimizing changes to the code.

- I. A must be constructed (using an appropriate call to the ArrayList class constructor) before any "work" with the ArrayList is attempted.
 - II. A call to the ArrayList's destructor should be added, to destroy A when it is no longer needed.
 - III. All references to elements of A via use of an index (e.g., A[I]) must be replaced with calls to appropriate ArrayList methods.
 - IV. If the array stores a primitive type, the values stored in the array must be changed, either explicitly or implicitly via autoboxing, to objects.
- A. I and II only
 - B. I, III and IV only
 - C. II and III only
 - D. I and IV only
 - E. I, II, III and IV

23. Does Java do boundary checking of arrays during program execution?

- A. Yes, always
- B. Yes, but only if a project file setting is made to enable such checking
- C. Yes, but it is limited to single-dimension arrays
- D. Yes, but only if the array is a member of a class
- E. No

When answering the next four questions, use this program fragment. `intObj` is a class containing one private field, an integer. `int getInt()` returns the value of this integer; `void setInt(int newValue)` changes its value to `newValue`.

```
intObj p;  
p = new intObj();  
intObj q = new intObj();  
q.setInt(20);  
p.setInt(q.getInt());           //line 1  
p.setInt(15);  
q = p;                           //line 2  
p = null;  
System.out.print(p.getInt());    //line 3
```

Use these responses when answering the next two questions:

	<u>value stored where p references</u>	<u>value stored where q references</u>
A.	20	20
B.	15	15
C.	20	15
D.	none—reference of null pointer	20
E.	none—reference of null pointer	none—reference of null pointer

24. What are the values stored in the locations that `p` and `q` reference *after* the line labeled `//line 1` is executed?

25. What are the values stored in the locations that `p` and `q` reference *after* the line labeled `//line 2` is executed?

26. What occurs when `//line 3` is executed?

- A. an exception is thrown
- B. the number 20 prints
- C. the number 15 prints
- D. the number 0 prints
- E. the number stored where `q` is referencing is printed

Consider this recursive function when answering the next 2 questions:

```
//intent:   if n is odd and positive, returns 1 + 3 + 5 + ... + n  
//         if n is even and positive, returns 2 + 4 + 6 + ... + n  
//         if n is 0 or negative, returns 0  
  
public static long addThemUp(int n)  
{   if (n > 0)  
    return n + addThemUp(n - 2);  
    else  
    return n;  
}
```

27. Which of these situations would cause `addThemUp` to return incorrect results?

- I. the parameter's value was negative
- II. the parameter's value was zero
- III. the parameter's value was positive and even
- IV. the parameter's value was positive and odd

- A. I only
- B. I and II only
- C. II and III only
- D. I and IV only
- E. I, II, III and IV

28. Which of these versions of `addThemUp` would return "as intended" results?

- A.

```
public static long addThemUp(int n)
{
    if (n >= 0)
        return n + addThemUp(n - 2);
    else
        return 0;
}
```
- B.

```
public static long addThemUp(int n)
{
    if (n > 0)
        return n + addThemUp(n - 2);
    else
        return 0;
}
```
- C.

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

- D. Both A and B
- E. A, B and C

29. Suppose you wanted to use a function to initialize three (already declared) variables that are not fields of a class. Further suppose you are doing this in a language that allows passing parameters either by value or by reference. To initialize the variables,

- A. you should pass them to the function by reference and set the values in the function; the variables will be initialized when the function returns.
- B. you should pass them to the function by value and set the values in the function; the variables will be initialized when the function returns.
- C. you should pass them to the function by value, set the values in the function, then return these values via a `return` statement.
- D. define these variables so they are global to the function, then set their values within the function; that is the preferred approach.
- E. you should adopt another approach: none of the above methods is a reasonable approach.

30. Here is a list of steps to be taken when a function with a parameter is called:

- I. execute the function (and use the formal parameter during its execution)
- II. delete the temporary storage associated with the formal parameter
- III. do whatever work is necessary to determine the actual parameter's value
- IV. do whatever work is necessary to resolve the actual parameter to a location in memory
- V. create temporary storage with the name of the formal parameter
- VI. copy the value of the actual parameter to the temporary storage location
- VII. copy the memory location of the actual parameter to the temporary storage location

Now suppose a function is called with a parameter that is passed by reference. *In order*, what steps are taken to execute the function, from the start of the call through its completion, that involve this parameter? (The lists below give steps in order when read left to right.)

- A. III, V, VI, I, II D. V, VI, IV, II, I
B. IV, V, VII, I, II E. IV, V, VI, II, I
C. V, VII, III, II, I

31. What is the output from the following Java program fragment?
(Reponses are on the next page.)

```
public static void main(String[] args)
{
    int A = 10;
    int B = 20;
    update(A, B);
    System.out.println(A + " " + B);
}

public static void update (int X, int Y)
{
    X = X + Y;
    Y = Y + X;
    System.out.println(X + " " + Y);
}
```

- A. None; there is a run-time error (before any output is produced)
- B. 10 20 D. 30 50
 10 20 10 20
- C. 10 20 E. 30 50
 30 50 30 50

32. Suppose you have a class `MyClass` and want to easily replace the contents of one object, `target`, with the contents of another object of `MyClass`, `source`. Which of the following statements would correctly create the copy?

- A. `target = source;`
- B. `target.clone(source);`
- C. `target = source.clone();`
- D. `target = source.equals();`
- E. `target = (MyClass) source.clone();`

33. To enable exception handling on a block of code, one

- A. encloses it in a `try` block
- B. encloses it in a `throwable` block
- C. labels it, then inserts that label into the Java exception handler list
- D. must place the code into its own method and mark the method `throwable`
- E. enables a `trap` for that code

34. In Java, what happens if code is written that could throw a checked exception in a method that has no `throws` clause for that exception, and there is no catch block defined in the method to handle that particular exception class?

- A. If an accessible catch block exists for one of the exception's ancestor classes, then the program compiles and runs.
- B. If there is no catch block that can handle the exception, the code will not compile.
- C. If the exception is thrown, and there is no catch block that can handle the exception, the program halts.
- D. If the exception is thrown, and there is no catch block that can handle the exception, the program continues, but its results are unpredictable
- E. Both A and B, taken together, fully describe what occurs.

35. A binary file opened for input only and relative access can

- I. have its information changed
 - II. have its file pointer moved sequentially through the file
 - III. have its file pointer positioned to any object stored in the file, regardless of where the file pointer was pointing just previously
 - IV. be checked for end-of-line conditions
- A. I and III only C. II and IV only E. I, III and IV only
B. II and III only D. I, II and IV only

Use this information when answering the next three questions:

Suppose I wanted to copy an `ArrayList<string>`'s contents into a file, with the string stored in a position of the `ArrayList` comprising one line of the file. The string in the first position of the `ArrayList` is to be the first line of the file, the string in the next position of the array is to be the next line of the file, and so on.

Here's an outline of what needs to be done, with two actions missing:

```
set current position of string array to its start;
_____ ; //missing action 1
while (items remaining in the array)
{
    put current position's string into the file
    (followed by the new line character);
    _____ ; //missing action 2
}
```

36. Which phrase best describes what missing action 1 should be?

- A. open the file as text, input, sequential access
- B. open the file as text, output, sequential access
- C. open the file as binary, output, relative access
- D. open the file as binary, output, sequential access
- E. open the file as text, input, relative access

37. Which phrase best describes what missing action 2 should be?

- A. go to the next position in the `ArrayList`
- B. go to the next line of the file (since the file pointer does not move automatically)
- C. go to the next component of the file (since the file pointer does not move automatically)
- D. Both A and B
- E. Both A and C

38. What is the output of this program fragment? (Ignore any leading spaces.)

```
double x = 123.321;
String Y = "Hi!";
System.out.format("%7.3f%s", x, Y);
```

- A. 23.32Hi!
- B. 123.321Hi!
- C. +23.32Hi!
- D. 1.23e2Hi!
- E. none; a compile- or run-time error occurs

39. When using `hasNext()` on a scanner, how do we know when the end of file has been reached?

- A. `hasNext ()` throws an `EOFException`.
- B. `hasNext ()` returns a `null`.
- C. Control transfers to the `finally` block associated with the block that `hasNext ()` is in
- D. `Scanner()` throws an `EOFException`.
- E. `hasNext ()` returns a `false`.

40. If a class is not qualified as `public` or `private`, what does that imply about its public methods and fields?

- A. No other class can use them: class access defaults to `private`.
- B. Only classes in the same file can use them.
- C. Only classes in the same package can use them.
- D. Only classes derived from this class can use them.
- E. All classes can use them.

41. What's the connection between `public` classes and `.java` file names? (Assume the file contains no inner classes.)

- A. The file can have at most one `public` class; if present, it must have the same name as the file.
- B. The file must have at least one `public` class and none of them can have the same name as the file.
- C. It is common practice to name a file after one of the `public` classes in it, but Java does not require it.
- D. If the file does not contain a `public` class, its name must not match the names of any of the classes in it.
- E. There is no connection!

42. An interface must meet which of these restrictions?

- I. It must not have any fields.
- II. All methods must be abstract.
- III. Only `public` methods are allowed.
- IV. A class can implement only one interface.

- A. II and III only
- B. I, II and III only
- C. II, III and IV only
- D. I, III and IV only
- E. I, II, III and IV

Use this information when answering the next seven questions:

`Polygon` is a class that defines regular polygons (figures such as equilateral triangles, squares, and regular pentagons—polygons where all the sides have the same length). It has, among other public methods, one named `area()`, which takes no parameters, and returns as type `double` the area of the polygon.

Classes `Square`, `EquiTriangle` and `Pentagon` are derived from `Polygon`. `Square` and `EquiTriangle` each have, among other public methods, one named `area()`, which takes no parameters and returns as type `double` the area of a `Square` and `EquiTriangle`, respectively. `Pentagon` does *not* define a method named `area()`.

43. What fields does `square` inherit from `Polygon`?

- A. Only those that have public or package access
- B. All except those that are private
- C. Only public ones
- D. All except those that have protected access
- E. All of them

44. Which among the classes `square`, `EquiTriangle` and `Pentagon` inherit `area()` from `Polygon`?

- A. `square`, `EquiTriangle` and `Pentagon`
- B. `square` and `EquiTriangle` only
- C. `Pentagon` only
- D. `square` only
- E. none of them

45. Define `P` to be a `Polygon` object reference, `s` a `square` object reference and `T` an `EquiTriangle` object reference. Which of these assignment statements are legal?

- | | |
|------------------------|-------------------------|
| I. <code>s = T</code> | III. <code>T = P</code> |
| II. <code>P = T</code> | IV. <code>P = s</code> |

- A. II and IV only
- B. II and III only
- C. II, III and IV only
- D. I, II and IV only
- E. I, II, III and IV

46. Suppose the object referenced in a cell of `ArrayList<Polygon>` `pentagonGroup` is of type `Pentagon`. If you later reference that object, what type will it have?

- A. `Pentagon`
- B. `Polygon`
- C. `Object`
- D. `Byte`
- E. `Class`

47. Suppose you wish to access the pentagon stored at position 3 of an `ArrayList<Pentagon> pentagonGroup` and invoke the `area()` method on it. Which statement below does this correctly?

- A. `pentagonGroup.get(3).area();`
- B. `pentagonGroup.area().get(3);`
- C. `pentagonGroup.(Pentagon).get(3).area();`
- D. `pentagonGroup[3].area();`
- E. `(Pentagon)pentagonGroup[3].area();`

48. Suppose you wish to call `Polygon`'s `area()` method in the definition of `EquiTriangle`'s `area()` method; both `area()` methods have the same signature. How is this done?

- A. `(polygon)area();`
- B. `super.area();`
- C. `this.area();`
- D. `area();`
- E. `parent.area();`

49. Suppose `Polygon`'s `area()` function is made abstract. Which of the following statements are then true?

- I. `Polygon` must be made an abstract class.
 - II. `Polygon`'s `area` function must have no body.
 - III. `Polygon` objects cannot be constructed.
 - IV. Any class derived from `Polygon` must override the `area()` function (if we want to be able to construct objects of those classes)
- A. I and III only
 - B. I and II only
 - C. I and IV only
 - D. I, III and IV only
 - E. I, II, III and IV

50. Which of the following statements about applets are true?

- I. An applet is a class that must be derived from the Java `Applet` class
 - II. An applet is invoked from HTML statements, rather than a "main()"
 - III. An applet almost always overrides the `paint()` method
 - IV. An applet must be compiled into a `.class` files before it can be used
- A. I and II only
 - B. II and III only
 - C. I, II and III only
 - D. II, III and IV only
 - E. I, II, III and IV

51. When is the `paint()` method called?

- A. Whenever the HTML of a Web page invokes it
- B. Whenever the Web browser determines the screen needs to be repainted
- C. Only when the user enters the page that executes the applet
- D. Whenever a geometric figure is created or modified
- E. Only when the applet is first invoked

52. The graphics window in Java has its origin in a specified position, and has the x and y coordinates' values increasing in specified directions. Which of the selections below correctly gives these specifications?

<u>origin is located</u>	<u>x value increase</u>	<u>y values increase</u>
A. upper left	to the right	up
B. upper left	to the right	down
C. upper left	to the left	up
D. center	to the right	up
E. center	to the right	down

53. To “notice” an event (and react to it), Java requires you to do which of the following things?

- I. define your own “handler” class that implements the appropriate Java-provided listener interface
 - II. define your own “handler” class that extends the appropriate Java-provided adapter class
 - III. construct an object of your “handler” class
 - IV. add your handler object to the list of listeners Java maintains
- A. Either I or II, and III and IV
 - B. I, II, III and IV
 - C. II and III only
 - D. I and III only
 - E. Either I or II, and IV only