CS 102 - Introduction to Programming
Midterm Exam #2 - Prof. Reed
Spring 2008

What is your name?: ___________________________(2 points)

There are three sections:
   I. True/False. . . . . . . . . . . . . . .54 points; (27 questions, 2 points each)
   II. Multiple Choice . . . . . . . .44 points; (11 questions, 4 points each)

---------------
98 points + 2 for name = 100 points total

This test is worth 15% of your final grade. You must put your answers on the bubble form. All code is in Java unless stated otherwise. This test is open book and open notes. You have 50 minutes.

• For the True/False questions, if the answer is True, fill in the A bubble on your answer form. If the answer is False, fill in the B bubble on your answer form.
• For the multiple choice problems, select the best answer for each one and select the appropriate letter on your answer sheet.
• Be careful - more than one answer may seem to be correct. Some questions are tricky.

I. True/False: (2 points each)

T  F  1. According to our text, the name of a Java class (e.g. Student) should be the same as the filename (e.g. Student.java)
T  F  2. A program can have more than one Constructor.
T  F  3. One way to tell the difference between a Constructor and a method is that a Constructor has no parameters.
T  F  4. The name of a Constructor must be the same as the class name unless you declare it as type Constructor.
T  F  5. You can have Constructors with the same name as a method name, but you must leave off the return value and declare it as type Constructor.
T  F  6. Overloaded methods can have the same name, but must have different numbers of parameters.
T  F  7. Methods can have the same name and number of parameters, as long as the types of the parameters are different in at least one case.
T  F  8. Methods can have the same name, number and type of parameters, as long as the return type of the methods are different.
T  F  9. If you don’t implement the toString() method in a class, then using an instance of that class in a System.out.println(...) statement displays a number.
T  F  10. Every method must have the return type declared unless the method doesn’t return anything, or the return value is of type boolean.
T  F  11. A static method can not access non-static instance variables.
12. A method can call itself in Java. [T] [F]  

13. If method A( ) calls method B( ) in Java, and method B( ) calls method C( ), then method C( ) may not call methods A( ) or B( ), since that would create an endless loop. [T] [F]  

14. The `length()` method is used for strings however the `length` property is used for arrays. [T] [F]  

15. Every set of `if-else-if` statements can be alternatively represented using a `switch-case` statement. Not if the `if-else-if` is comparing Strings or double values. [T] [F]  

16. The output of the following code in Java is:  

```java  
int x, answer;  
for(x=1, answer=0; x<=100; x++) {  
    answer = answer + x;  
}  
System.out.println( answer);  
```  

17. The output of the code below is the value: 13  

```java  
int x=3, y=5, z=2;  
System.out.println(x + y * z);  
```  

18. The output of the following lines of code is: No  

```java  
boolean False = false;  
if (False = true) {  
    System.out.println("Yes");  
} else {  
    System.out.println("No");  
}  
```  

19. The following code is valid (compiles and runs) in Java:  

```java  
for( ; ; )  
; // empty line  
```  

20. The following `if` statement:  

```java  
if( a<b)  
    if( c<d)  
        if( e<f)  
            answer = e;  
```  

can equivalently be written as:  

```java  
if( (a<b) && (c<d) && (e<f) )  
    answer = e;  
```
21. The output of the program segment below is the text: **First One ***

```java
boolean value = false;
if (value = true) // assignment statement again
    System.out.print("First One");
else
    System.out.print("Second One");
System.out.println(" *** ");
```

22. The following code in Java stores in variable `sum` the sum of positive integers less than 10:

```java
int sum = -1;
for(int x = ++sum; x < 10; x++)
    sum += x;
```

23. The following code prints the words: **Grade is: B**

```java
int x = 87;
char result = 'A';
System.out.print("Grade is: ");
if (x > 90)
    result = 'A';
if (x > 80)
    result = 'B';
if (x > 70)
    result = 'C';
if (x > 60)
    result = 'D';
System.out.print(result);
```

24. The following code prints the words: **score is: 85**

```java
char val = 'A';
int score = 85;
switch (val){
    case ('A'): score = 95; break;
    case ('B'): score = 85; break;
    case ('C'): score = 75; break;
}
System.out.println("score is: " + score);
```
25. Assume the code shown below, where method `first()` is called.
Output of this segment of code is: Values are: 8 3

```java
class SwapThem
{
    int x = 3;
    int y = 8;

    void first()
    {
        swapValues( x, y);
        System.out.println("Values are: " + x + " " + y);
    } // end method first()

    void swapValues(int a, int b)
    {
        int temp = x; // these are ignored in this method. Instead, the instance
        x = y; // variables (global within this file) are used.
        y = temp;
    } // end method swapValues()
}
//end class SwapThem()
```

26. Assume the code shown below, where method `swapValues1` is called.
Output of this segment of code is: Values are: 2 3

```java
int[] numbers = {1,3,2,4};
swapValues1(numbers[1], numbers[2]);
System.out.println("Values are: " + numbers[1] + " " + numbers[2]);
// ... other code
public void swapValues1(int num1, int num2)
{
    int temp = num1;
    num1 = num2;
    num2 = temp;
}
```

27. Assume the code shown below, where method `swapValues2` is called.
Output of this segment of code is: Values are: 2 3

```java
int[] numbers = {1,3,2,4};
swapValues2(numbers, 1, 2);
System.out.println("Values are: " + numbers[1] + " " + numbers[2]);
// ... other code
public void swapValues2(int[] numbers, int i, int j)
{
    int temp = numbers[i];
    numbers[i] = numbers[j];
    numbers[j] = temp;
}
```
II. Multiple Choice (4 pts. each)

28. Consider the program segment given below. Its output is:

```java
int x = 2;
int y = 3;
String z = "";
System.out.println(x + y + z + " is the answer");
```

a) 2 + 3 is the answer  
b) 23 is the answer  
c) 5 is the answer  
d) 2 3 is the answer  
e) None of the above

29. Consider the code given below. Its output is:

```java
int x = 3;
int y = 6;
int z = x++ + y;
System.out.println("Value is: "+x+y+z);
```

a) Value is: 369  
b) Value is: 469  
c) Value is: 3710  
d) Value is: 4610  
e) None of the above

30. Consider the code given below. If its output is:

```
9   12   15  
12   16   20  
15   20   25  
18   24   30
```

what are the values for variables start, end, first and last?

```java
for( int i=start; i<=end; i++) {
    for( int j=first; j<last; j++) {
        System.out.printf("%5d",i*j);
    }
    System.out.println();
}
```

a) int start=3, end=6, first=3, last=6;  
b) int start=1, end=4, first=9, last=18;  
c) int start=9, end=4, first=9, last=3;  
d) int start=3, end=4, first=9, last=18;  
e) None of the above
31. What is the output of the code below if we call the code using:

```java
Compare compareInstance = new Compare(4);
```

```java
class Compare {
    int x=4, y=2;
    Compare( int x)
    {
        this.x = x;
        first();
    }

    void first()
    {
        int n=2;
        if( second(n,x) && second(y,n))
            System.out.print(x + "," + y);
        System.out.println("Done");
    }

    boolean second( int a, int b)
    {
        x = b/a;
        y = y-x;
        if( x<3)
            return true;
        else
            return false;
    }
}
```

Options:

- a) 4, 2 Done
- b) 2, 0 Done
- c) 0, 2 Done
- d) Done
- e) None of the above

32. Again consider the code above. What is the output of the code above if we call the code using:

```java
Compare compareInstance = new Compare(8);
```

```java
class Compare {
    int x=4, y=2;
    Compare( int x)
    {
        this.x = x;
        first();
    }

    void first()
    {
        int n=2;
        if( second(n,x) && second(y,n))
            System.out.print(x + "," + y);
        System.out.println("Done");
    }

    boolean second( int a, int b)
    {
        x = b/a;
        y = y-x;
        if( x<3)
            return true;
        else
            return false;
    }
}
```

Options:

- a) 4, 2 Done
- b) 2, 0 Done
- c) 0, 2 Done
- d) Done
- e) None of the above
33. Consider the program method given below.

```java
public void handleArray( int[] theNumbers) {
    int x = theNumbers[0];
    for (int i=0; i<theNumbers.length; i++) {
        if (theNumbers[i] < x)
            x = theNumbers[i];
    }
    System.out.println( x);
}
```

Its output is:

a) The minimum value in array `theNumbers`
b) The maximum value in array `theNumbers`
c) The average value in array `theNumbers`
d) The sum of the values in array `theNumbers`
e) None of the above

34. Assume you are doing a numbers guessing game, where you are attempting to guess a number between 500 and 750 (this is a binary search). After each guess, you are told if you need to guess higher or lower. What is the maximum number of guesses you would need to find the number?

a) 8
b) 9
c) 10
d) 11
e) None of the above

35. What is the output of the code given below when `problem35Driver()` is called?

```java
public void handleArray( int[] theNumbers) {
    int x = theNumbers[0];
    for (int i=0; i<theNumbers.length; i++) {
        if (theNumbers[i] < x)
            x = theNumbers[i];
    }
    System.out.println( x);
}
```

```java
public void problem35Driver() {
    char[] theArray = {'H','e','r','e','','w','e',' ','g','o'};
    problem35( theArray);
    // display array contents
    for( int i=0; i<theArray.length; i++)
        System.out.print( theArray[i]);
    System.out.println();
}
```

```java
public void problem35(char[] w) {
    int x = w.length;
    char c;
    for (int i=0; i<x/2; i++) {
        c = w[i];
        w[i] = w[x-i-1];
        w[x-i-1] = c;
    }
}
```

a) The contents of the original array in reverse order
b) The contents of the original array in the original order
c) The contents of the original array with half of the characters reversed
d) The original array with characters rearranged so they are neither in the original nor reversed order
e) None of the above
36. What is the output of the program segment at right below when an instance of class Verify is created and used to call method verifyIt()?

   a) -1  
   b) 0  
   c) 1  
   d) 2  
   e) None of the above

37. What is the return value of the call: methodB(2,5);

   a) 7  
   b) 10  
   c) 25  
   d) 32  
   e) None of the above
38. What is the output of the code given in the two columns below when an instance of class `Confuse` is created and used to call method `startUp()`?

```
class Confuse {
    int x=2;
    int y=5;

    private void first(int y, int x) {
        x += 1; y += 2;
    }

    private void second(int x, int s) {
        x += 3; y = 7;
    }

    private void third(int y) {
        setXY( y);
        x += 2; y += 3;
    }

    private void setXY( int y) {
        x = y;
        this.y = x-1;
        second(y, x);
    }

    private void display() {
        System.out.println(x + y);
    }

    public void startUp() {
        int x=1, y=2;
        first(x,y);
        second(x,y);
        third( 4);
        display();
    }
}
```

a) 9
b) 11
c) 13
d) 15
e) None of the above