

CS 102 - Introduction to Programming
Midterm Exam #1 - Prof. Reed
Fall 2009

What is your name?: _____

There are two sections:

I. True/False. 60 points; (30 questions, 2 points each)

II. Multiple Choice 40 points; (8 questions, 5 points each)

100 points total

This test is worth **10%** of your final grade. Please fill in your answers on the bubble form. After the test you may keep these pages, but you must turn in your bubble form. This test is open book and open notes. You have 50 minutes.

- 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. Many questions are tricky.
- Some problems ask you to determine whether something is *valid*. Something is *valid* if it would not generate a compiler error and would execute without the program crashing.

I. True/False: (2 points each)

T F 1. The same valid Java program can be run either from within BlueJ or from the command line.

T F 2. A java class name should be the same as the filename in which it is found. For instance the *Book* class should be stored in *Book.java*

T F 3. A single line comment using // can be safely nested inside a block style comment that uses /*...*/

T F 4. Multiple variables can be declared and initialized at one time, such as:
`int x=1,y=2,z=3;`

T F 5. Multiple variables can all be initialized on the same line, such as:
`x=y=z=0;`

T F 6. The output of the statements below is: 3

```
int value = 7;
System.out.println( value / 2);
```

T F 7. The output of the statements below is: 3

```
int value = 7;
System.out.println( (int)value / 2);
```

T F 8. The output of the statements below is: 3

```
int value = 7;
System.out.println( (double)1 * value / 2);
```

- T F 9.** The output of the statements below is: 3
- ```
int value = 7;
System.out.println(value / 2 * 1.0);
```
- T F 10.** The output of the statements below is: 3
- ```
double value = 7.0;
System.out.println( value / 2);
```
- T F 11.** The following statements are *valid* (compile and run) in Java:
- ```
int x = 0;
x+= 1;
```
- T F 12.** The following statements are *valid* (compile and run) in Java:
- ```
int x = 3;
int y = 5
x += x+++y++;
```
- T F 13.** It is possible in Java to write one line of code that generates 2 lines of output.
- T F 14.** It is possible in Java to write 2 lines of code that generates 1 line of output.
- T F 15.** Multi-word variable names in Java should have the first letter of each word capitalized, except for the first word.
- T F 16.** When creating an instance of the Scanner class, any valid identifier can be used.
- T F 17.** When writing code to select a menu option declared as a *char*, a *do* loop is better than a *while* loop.
- T F 18.** When writing code to select a menu option declared as a *char*, a *switch-case* statement is preferable to multiple *if-else-if* statements.
- T F 19.** When writing code that repeats several lines 0 or more times, a *do* loop is preferable to a *while* loop.
- T F 20.** When writing code that repeats several lines 0 or more times, a *do* loop is preferable to a *for* loop.
- T F 21.** A Java program could have all indentation removed and it would still give exactly the same output;
- T F 22.** The following statements compile and run in Java:
- ```
String name = "One";
if(name.length() = 3) {
 System.out.println("Are equal");
}
```
- T F 23.** To check and see if the value of variable x is between 3 and 9 we could use:.
- ```
int x = 4;
if( 3 < x < 9) {
    System.out.println("x is between 3 and 9");
}
```

T F 24. The output of the statement below is: 2

```
System.out.println( -7 % -3);
```

T F 25. After running the code shown below, the value stored in variable `x` is: 12

```
int y = 3;
int x = y * y++;
```

T F 26. The output of the following lines of code is: Done End

```
boolean notDone = false;
if (notDone = true)
    System.out.println("not Done ");
if( notDone = false)
    System.out.println("Done ");
else
    System.out.println("Undecided ");
System.out.println("End");
```

T F 27. The output of the following statements is: 3 Done

```
int x = 2;
int y = 1;
System.out.print( x + y + "");
System.out.println(" Done");
```

T F 28. The output of the following statements is: 3 Done

```
int x = 2;
int y = 1;
System.out.print( "" + x + y);
System.out.println(" Done");
```

T F 29. The output of the program segment below is: 3 Done

```
String jumbo = "2";
String shrimp = "1";
System.out.println(jumbo + shrimp);
System.out.println(" Done");
```

T F 30. The following code prints the words: 12 Done

```
char c='b';
switch (c){
    case 'a': System.out.print("1");
    case 'b': System.out.print("2");
    break;
    case 'c': System.out.print("3");
    break;
}
System.out.println(" Done");
```

II. Multiple Choice (4 points each)

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

```
int i=1;
while (i<=100) {
    System.out.print(i);
    if( i%10 == 0) {
        System.out.println();
    }
    i++;
}
```

- a) All the numbers from 1 to 100, with a line break after the number 10
- b) All the numbers from 1 to 100 in a grid of 10 rows and 10 columns
- c) All the numbers from 1 to 101, 4 per row
- d) All the numbers from 1 to 101, with a line break after every 10 numbers
- e) None of the above

32. The output of the following code in Java is equal to:

```
int answer=2;
int limit = 5;
for(int x=0; x<limit; x++) {
    answer = answer + 1;
}
System.out.println( answer);
```

- a) $2 * 5$
- b) $2 + 5$
- c) 25
- d) 32
- e) None of the above

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

```
int answer=2;
int limit = 5;
for(int x=0; x<limit; x++) {
    answer = answer * 2;
}
System.out.println( answer);
```

- a) $2 * 5$
- b) $2 + 5$
- c) 25
- d) 32
- e) None of the above

34. Consider the code shown at right below, that uses the Circle class demonstrated during class. What does the output of this code look like?

<p>a) A circle that grows larger as it moves to the right.</p> <p>b) A circle that itself moves in a clockwise circle.</p> <p>c) A circle that itself moves in a counter-clockwise circle.</p> <p>d) A circle that moves in an outward growing clockwise spiral.</p> <p>e) None of the above.</p>	<pre>public class picture { public static void main() { Circle firstCircle = new Circle(); firstCircle.makeVisible(); for (int i=1;i<31;i++) { firstCircle.slowMoveVertical((-i)*2); firstCircle.slowMoveHorizontal((-i)*2); firstCircle.slowMoveVertical(i*2); firstCircle.slowMoveHorizontal(i*2); }//end for(int i... }//end main... } //end class picture</pre>
---	---

35. Consider the method given below. It can best be described as a method that:

```
public boolean method35( int number)
{
    if ( (number==1) || (number==2) ) {
        return true;
    }

    for (int i=2; i < number/2; i++) {
        if ( number%i == 0) {
            return false;
        }
    }
    return true;
}
```

- a) checks to see if a number is even
- b) checks to see if a number is odd
- c) checks to see if a number is prime
- d) does not compile
- e) None of the above

36. Consider the method given below. If its output is:

```
3   6   9
4   8  12
5  10  15
6  12  18
```

How was the method called?

```
public void methodB(int r, int k, int n)
{
    int i, j;

    for( j=k; j<=n; j++) {
        for( i=1; i<=r; i++) {
            System.out.print("   " + i*j);
        }
        System.out.println();
    }
}
```

- a) methodB(3, 3, 1);
- b) methodB(4, 1, 3);
- c) methodB(3, 6, 3);
- d) methodB(4, 6, 3);
- e) None of the above

37. Consider the method given below. What would be the output if this were called using the statement:

```
System.out.println( method37( 54321));
```

```
int method37( int number)
{
    int x = number;
    int y = 0;
    while ( x > 0) {
        y = y * 10 + x%10;
        x = x / 10;
    }
    return y;
}
```

- a) 5
- b) 12345
- c) 234
- d) 1
- e) None of the above

38. Consider the method given below. What would be the output if this were called using the statement:
method38();

```
void method38()  
{  
    int x = 13572;  
    while (x > 10) {  
        x = x / 10;  
    }  
    System.out.print( x);  
}
```

- a) 2
- b) 27531
- c) 1357
- d) 1
- e) None of the above