

Graded: Circle the bullet in front of each statement that accurately describes your instructor's course policies:

- Having a copy of all or part of another student's source code, just to look at for a little help, is considered cheating.
- You must submit every lab assignment in order to pass this course.
- The final exam for this course will be comprehensive.
- If you miss more than three lectures, your grade will be dropped by half a letter grade.
- If you don't do your homework you will not be allowed to attend some classes.
- All students must submit a copy of their lecture notes before the end of the day.
- Lab attendance is required.
- If you submit an assignment 45 minutes after the due date, it will not be considered late.
- Your instructor will give you a free lunch if you invite him to lunch.

Do not circle the bullet in front of any statement that incorrectly describes your instructor's course policies. Feel free to provide an explanation for your answer if you believe the statement to be ambiguous.

Not Graded: How much time do you plan to spend on this class outside of class?

Do you have that time scheduled on your calendar?

From what you have seen so far, do you have any suggestions for things that I could do better?

(a) (6 points) Complete the following program that should ask the user to enter two integers and display the sum (addition) of the two numbers.

```
public static void main(String[] ignored) {  
    System.out.println("Enter an integer value");  
    Scanner in = new Scanner(System.in);  
    int a = in.nextInt();  
}
```

(b) (2 points) What will be stored in the variable x:

```
double x = 3/6;
```

(c) (2 points) What will be stored in the variable x:

```
double x = 7 + 8 * 2;
```

Consider the following program:

```
public static void main(String[] args) {
    Scanner in = new Scanner(System.in);
    System.out.println("Please enter an integer.");
    int num = in.nextInt();
    if(num > 0) {
        System.out.println("A");
        if(num < 3) {
            System.out.println("B");
        } else {
            while(num > 0) {
                System.out.println("C");
                num -= 2;
            }
        }
    }
    if(num == -3) {
        System.out.println("D");
    } else {
        System.out.println("E");
    }
}
```

(a) What will be displayed if **13** is entered by the user?

(b) What will be displayed if **3** is entered by the user?

(c) What will be displayed if **-3** is entered by the user?

1. (3 points) Determine which of the following is the correct way to convert a double to a String and explain why:

```
double pi = Math.PI;  
String number = Double.toString(pi);
```

or

```
Double pi = new Double(Math.PI);  
String number = pi.toString();
```

2. (7 points) Write a program that asks the user to enter a phrase and then displays the number of spaces (only count spaces) entered.

```
public class Quiz4 {  
    public static void main(String[] ignored) {
```

(a) Implement a method for subtracting one complex number from another complex number. The method should be called `minus`.

(b) Draw the memory diagram for all of the variables visible from within the `minus` method just prior to returning from the method.

True/false

- _____ The top-down design methodology is good because it keeps everyone focused on a common goal.
- _____ The top-down design methodology is good because it avoids “reinventing the wheel.”
- _____ The top-down design methodology is good because it minimizes the chances of solving the wrong problem.
- _____ In the interest of incapsulation, use local variables instead of instance variables whenever possible.

In a top-down design process, which do you decide on first — the classes or the `public` methods?

When should you use bottom-up design?

Write a class method called `quiz8` that accepts an array of integers called `nums` and returns how many values in `nums` that are evenly divisible by 3. For example, if passed 10, 3, 8, 12, 15, the method should return 3.

Write a class method called `quiz9` that accepts an array of integers called `nums` and returns an `ArrayList` of integers containing all values in `nums` that are evenly divisible by 3. For example, if passed 20, 3, 8, 12, 15, the method should return 3, 12, 15.