
COMP132 – Computer Science II
Fall 2015

Homework #1
Java Review

1. Write a program (i.e. a class with a main method) that takes the user's first and last names as command line arguments and displays a welcome message incorporating the user's first and last name. For example, if the program is run as:

```
java MyProgram Bob Smith
```

Then the output would be: `Hello Bob Smith.`

But if the program is run as:

```
java MyProgram Jane Jones
```

Then the output would be: `Hello Jane Jones.`

For this question you will need to create a new project in Eclipse, create a new class within that project and then create a main method in that class. You can find information about creating new projects and new classes in the How-To document linked to from the course home page. Copy and paste your main method as your solution to this question.

2. What happens when your program from question 1 is run and the user only enters their first name? For example:

```
java MyProgram Sue
```

Briefly explain why this happens.

3. State which Java primitive data type you would use to hold each of the following values:

- a. The population of the world.
- b. The number of students enrolled at Dickinson College.
- c. The number of atoms in the universe.
- d. The national debt of the United States in dollars and cents (e.g. 123.45).
- e. Daily high temperature in whole degrees Celsius in New York City.

4. Consider the following variable declarations (the `Candidate` class can be found in the `comp132.examples.review` package of the `132SampleCode` project):

```
Candidate c1 = new Candidate("Bob", Candidate.REPUBLICAN);
Candidate c2 = new Candidate("Sam", Candidate.DEMOCRAT);
Candidate c3 = new Candidate("Jane", Candidate.INDEPENDENT);
```

a. Given the above variable declarations, and without executing any code on a computer, give the output that would be produced by the following lines of code.

```
c1.increaseVotes();
c2.increaseVotes(2);
c3.increaseVotes(3);

System.out.println("c1: " + c1.getName() + " : " + c1.getParty() +
    " : " + c1.getVotes());
System.out.println("c2: " + c2.getName() + " : " + c2.getParty() +
    " : " + c2.getVotes());
System.out.println("c3: " + c3.getName() + " : " + c3.getParty() +
    " : " + c3.getVotes());
```

b. Assuming the variables given above are recreated (i.e. do not continue from part a), and without executing any code on a computer, give the output that would be produced by the following lines of code.

```
c1 = c3;
c1.increaseVotes();
c2.increaseVotes(2);
c3.increaseVotes(3);
System.out.println("c1: " + c1.getName() + " : " + c1.getParty() +
    " : " + c1.getVotes());
System.out.println("c2: " + c2.getName() + " : " + c2.getParty() +
    " : " + c2.getVotes());
System.out.println("c3: " + c3.getName() + " : " + c3.getParty() +
    " : " + c3.getVotes());
```

c. Assuming the variables given above are recreated (i.e. do not continue from part a or part b), and without executing any code on a computer, give the output that would be produced by the following lines of code.

```
c2 = c3;
c3 = c1;
c1.increaseVotes();
c2.increaseVotes(2);
c3.increaseVotes(3);
c2.setParty(Candidate.LIBERTARIAN);
System.out.println("c1: " + c1.getName() + " : " + c1.getParty() +
    " : " + c1.getVotes());
System.out.println("c2: " + c2.getName() + " : " + c2.getParty() +
    " : " + c2.getVotes());
System.out.println("c3: " + c3.getName() + " : " + c3.getParty() +
    " : " + c3.getVotes());
```

5. Consider the following class definition, which contains both overloaded constructors and overloaded methods:

```
public class OverloadingHW {  
  
    private int x;  
    private int y;  
  
    public OverloadingHW() {  
        this(3);  
    }  
  
    public OverloadingHW(int x) {  
        this(x, 2);  
    }  
  
    public OverloadingHW(int x, int y) {  
        this.x = x;  
        this.y = y;  
    }  
  
    public int getX() {  
        return x;  
    }  
  
    public int getY() {  
        return y;  
    }  
  
    public void foo() {  
        x = 2;  
        y = 4;  
    }  
  
    public void foo(int a) {  
        x = x + a;  
        y = 7;  
    }  
  
    public void foo(int a, int b) {  
        this.foo(a+b);  
    }  
}
```

a. With the above class definition, and without executing any code on a computer, give the output that would be produced by the following lines of code.

```
OverloadingHW ohw = new OverloadingHW(3,5);  
System.out.println("x=" + ohw.getX() + " y=" + ohw.getY());  
ohw.foo();  
System.out.println("x=" + ohw.getX() + " y=" + ohw.getY());
```

b. With the above class definition, and without executing any code on a computer, give the output that would be produced by the following lines of code.

```
OverloadingHW ohw = new OverloadingHW(4);  
System.out.println("x=" + ohw.getX() + " y=" + ohw.getY());  
ohw.foo(2);  
System.out.println("x=" + ohw.getX() + " y=" + ohw.getY());
```

c. With the above class definition, and without executing any code on a computer, give the output that would be produced by the following lines of code.

```
OverloadingHW ohw = new OverloadingHW();  
System.out.println("x=" + ohw.getX() + " y=" + ohw.getY());  
ohw.foo(2,6);  
System.out.println("x=" + ohw.getX() + " y=" + ohw.getY());
```