CS-207: Programming II
Spring 2016
Northeastern Illinois University
Homework #7: Due 03/10/16 at 9:00 a.m.
Inheritance/Polymorphism

Problem #1

Determine the **exact** output of the `main` method on the next page. You should do this by hand and check your output by coding it. Type the output in a file named HW7_Problem1.txt and put the file in a a folder named Homework7 (to be submitted to D2L).

```
public class Harry
{
    private int h;

    public Harry(int h)
    {
        System.out.println("Harry");
        this.h = h;
    }

    public int getH()
    {
        return this.h;
    }

    public void m1()
    {
        System.out.println("Harry1");
    }

    public void m2()
    {
        m1();
        System.out.println("Harry2");
    }

    public boolean equals(Object o)
    {
        Harry h1 = (Harry) o;
        System.out.println("Harry Equality");
        boolean e = false;
        if (this.h == h1.getH())
            e = true;
        return e;
    }
}
```

```
public class Larry extends Harry
{
    public static int c = 0;

    public Larry()
    {
        super(5);
        System.out.println("Larry");
        c += super.getH();
    }

    public void m1()
    {
        System.out.println("Larry1");
        super.m1();
        c++;
    }

    public boolean equals(Object o)
    {
        Harry h1 = (Harry) o;
        System.out.println("Larry Equality");
        boolean e = false;
        if (this.getH() != h1.getH())
            e = true;
        return e;
    }

}
```

```
public class Mary extends Larry
{
    public Mary()
    {
        System.out.println("Mary");
        c += 2;
    }

    public void m2()
    {
        System.out.println("Mary2");
        c += super.getH();
    }

    public void m3()
    {
        super.m1();
        System.out.println("Mary3");
    }
}
```

```
public class HarryMaryLarryTest
{
    public static void main(String[] args)
    {
        Mary m = new Mary();
        m.m3();
        Larry la = new Larry();
        System.out.println(Mary.c);
        System.out.println(Larry.c);

        Harry[] hs = new Harry[4];
        hs[0] = new Harry(2);
        hs[1] = m;
        hs[2] = new Harry(5);
        hs[3] = la;

        for (int i = 0; i < hs.length; i++)
        {
            Harry h = hs[i];
            h.m1();
            h.m2();
            System.out.println(Larry.c);
        }

        for (int i = 1; i < hs.length; i++)
        {
            boolean b = hs[i-1].equals(hs[i]);
            System.out.println(b);
        }
    }
}
```

Problem #2:
Create a properly encapsulated class named `Shoe` that has the following:

- A String instance variable named `brand`.
- A `double` instance variable named `size`.
- A `String` instance variable named `color`.
- A constructor that takes 3 parameters, a `String`, a `double`, and a `String` for `brand`, `size`, and `color` (in that order) and sets the instance variables.
- A method named `display` that does not take any parameters and does not return anything. It should print out the values of the instance variables in the following format (for example):
  `Brand: Nike`
  `Size: 7.5`
  `Color: Blue`

Create a properly encapsulated class named `Stiletto` that inherits from `Shoe` (don't forget the keyword needed for inheritance!!) that has the following:

- A `double` instance variable named `height`.
- A constructor that takes 4 parameters, a `String`, a `double`, a `String`, and a `double` for `brand`, `size`, `color` and `height` (in that order) and sets the instance variables. (Hint: `super`)
- Override the `display` method. The overridden method should call the `display` method from the superclass and then print out the `height` in the following format:
  `Height: 3.5`
- Download the `ShoeInheritanceTest.java` file and make sure all your files are in the same folder. If you created your classes correctly, `ShoeInheritanceTest` will produce the output below.
- Place the `Shoe.java` and `Stiletto.java` files in the Homework7 folder to be submitted to D2L.

```
Creating a regular shoe.
Brand: Reebok
Size: 10.0
Color: White

Creating a stiletto shoe.
Brand: Stuart Weitzman
Size: 6.0
Color: Black
Height: 4.5
```

Problem #3:
Create a properly encapsulated class named `Word` that has the following:

- A String instance variable named `word`.
- A constructor that takes 1 parameter, a `String` and sets the instance variable.
- A getter for the `word` instance variable.
- Override the `Object equals` method. The overridden method should first print out the text `"Word equality"`. Then the method should determine whether two `Word` objects are equal by checking if their `word` instance variables are equal. Remember that the `equals` method must take an `Object` as a parameter - you'll need to use casting! Also remember to use your getter method with the parameter that is passed in.

Create a properly encapsulated class named `ConsonantWord` that inherits from `Word` (don't forget the keyword needed for inheritance!!) that has the following:

- A `String` instance variable named `cWord`.
- A constructor that takes 1 parameter, a `String`. This parameter should be used to set the instance variable of the superclass (Hint: `super`). Then, set the `cWord` instance variable the value of the parameter with all of the vowels removed. You may **not** use any loops or conditionals to do this.
- A getter for the `cWord` instance variable.

- A public method named `isSubstring` that does not take any parameters and returns a `boolean`. This method should determine if `cWord` is a substring of the superclass instance variable `word`. You may `not` use any conditionals or loops to do this. Don't forget how to access methods from the superclass in the subclass (Hint: super!!). Since you do not know the value of the `cWord` instance variable until runtime, you will need to form your regex expression by concatenating it all together. For example, `".*" + this.varname` would form a regular expression that tries to match a pattern of any number/type of characters before the value stored in the varname variable and then the varname value..

- Download the `WordInheritanceTest.java` file and make sure all your files are in the same folder. If you created your classes correctly, `WordInheritanceTest` will produce the output below.

- Place the `Word.java` and `ConsonantWord.java` files in the Homework7 folder to be submitted to D2L.

```
New ConsonantWord object
Superclass word: apple
Subclass word: ppl
Is cWord a substring of word? true

New ConsonantWord object
Superclass word: banana
Subclass word: bnn
Is cWord a substring of word? false

New Word object
Instance variable word: apple
Word equality
Is ConsonantWord object cw1 equal to Word object w1? true
```

A note on cheating/plagiarism:
A plagiarism detector is used on all submitted code (across all sections) for homework assignments. If the plagiarism detector determines that 25% or more of your code for a particular assignment is plagiarized, you will receive a zero (i.e. an F) for that homework assignment, regardless of whether you cheated from someone or vice-versa. If you plagiarize half or more of the total homework assignments, you will receive a zero for the entire homework percentage.

Submitting your assignment to D2L

1. Make sure your name and assignment number are in the .java file(s) (as comments) and text file.

2. Place all your files in a folder and compress (i.e. .zip) the folder. Submit the .zip file to the Homework #4 folder on D2L. You should submit only one file - the .zip file. Do **NOT** upload multiple files.

3. Turn your homework in to D2L by the specified deadline (no late homework will be accepted - see syllabus for policies)