CS 163/164 – Classes, Objects, Loops Review

Name(s): ____

1. Analyze the class Square to answer the questions:
    a. Does it store any value/instance variable? Yes, side is an instance variable.
    b. List each method and explain what each method does.
       Constructor – call the setSide method that initializes the side instance variable with zero (0)
       setSide – receives a parameter and initializes the instance variable with the parameter value
       getSide – returns the current instance variable value
       calculateArea – return the area of the square
       toString – return a String containing the message "Side: " and the value of the instance variable side.
    c. Why the method setSide needs to use this.side? What "this" means in that context? Because the parameter has the same name of the instance variables. This means the instance variable of the object.

```
public class Square {
    private int side;
    public Square(){
        setSide(0);
    }
    public void setSide(int side){
        this.side = side;
    }
    public int getSide(){
        return side;
    }
    public int calculateArea(){
        return side * side;
    }
    public String toString(){
        return "Side: " + side;
    }
}
```

2. Analyze the class AppSquare to answer the questions:
    a. Does it store any value/instance variables? No, there is no private variables declared outside of the main method.
    b. What the main method is doing? – see the comments in the code below
    c. Where the setSide method from Square is being called? Why is this called like that? setSide is being called after we read a value for the side variable, setSide receives a parameter so you sent that parameter. To call methods over objects we need to use nameOfTheObject.nameOfTheMethod(<parameters if they exist>)
    d. Where the getSide method from Square is being called? Why is this called like that? getSide is being called everytime we want to print the value of the instance variable for sq1. It is being called inside of a System.out.println because it return a value that will be printed in the terminal.

```
import java.util.Scanner;

public class AppSquare {
    public static void main(String args[]){
        Square sq1 = new Square(); //creates a Square object of name sq1
        Scanner in = new Scanner(System.in); //creates a Scanner object of name in
        System.out.println("Do you want to run the program (y/n)"); //print a message
in the terminal
        char answer = in.nextLine().charAt(0);//read a String and get the first
character from that String - that is the way to read char in Java
        while(answer != 'n'){ //repeats the code while the user did not type 'n'
            System.out.println("Enter a side:"); //print a message in the terminal
            int side = in.nextInt(); //reads an int value and stores in the side
variable
            System.out.println("Current side of the square: " + sq1.getSide());
//print a message and the return of the getSide method from sq1 object
            System.out.println("Changing the side of the Square:"); //prints a message
in the terminal
            sq1.setSide(side);//change the side of the sq1 object
            System.out.println("New side of the square: " + sq1.getSide()); //prints a
message and the side of the sq1 object that is returned from getSide method
            int area = sq1.calculateArea(); //call the calculateArea for sq1 object
and stores the returned value in the area variable
            System.out.println("Area : " + area + " " + sq1.toString()); //prints
message followed by the area and the String that is returned from the toString called
over sq1 object
            System.out.println("Do you want to run the program (y/n)");//prints a
message in the terminal
            in.nextLine(); //need to have this so the next String reading is not
skipped
            answer = in.nextLine().charAt(0); //reads a String and get the first char
from the String that was read
        }
    }
}
```

3. Implement the method numberOfCharInString and do a call to that method in the main method.

```
import java.util.Scanner;
public class AppStrings {
    public static void main(String args[]){
        Scanner in = new Scanner(System.in);
        System.out.print("Enter a string: ");
        String msg = in.nextLine();
        System.out.println("Enter a char:");
        char c = in.nextLine().charAt(0);
        //call the method numberOfCharInString
        int times = numberOfCharInString(c, msg);
        System.out.printf("Number of times that %c appears in %s is %d\n", c, msg,
times);
    }
    /**
     * numberOfCharInString
     * Method finds and return the number of times
     * that char c is in String str
     * @param c
     * @param str
     * @return int
     */
    public static int numberOfCharInString(char c, String str){
        int time = 0;
        for(int I = 0; i < str.length(); i++){
             char c1 = str.charAt(i);
              if(c == c1) time++;
        }
        return time;
    }
}
```

4. Write the exact output of this program.

```
public class StringFormatProgram {
   public static void main(String[] args) {
      int value1 = 12;
      double value2 = 25.75986;
      String color = "red";
      int len = color.length();
      System.out.println("Length of color: " + len);
      System.out.println("First letter of color: " + color.charAt(0));
      System.out.printf("%d\n", value1);
      System.out.printf("%.2f\n", value2);
      System.out.printf("%20.2f\n", value2);
      System.out.printf("%s\n", color);
      System.out.printf("%S\n", color);
   }
}
```

```
Length of color: 3
First letter of color: r
12
25.76
               25.76
red
RED
```