CIS 201 – Computer Science I
Laboratory Assignment 7

Introduction

In this lab you will work with loops.

Create a directory, `Lab07` in your `CS1` directory. Do all of your work in this the `Lab07` directory.

A `for` loop has the following syntax:

```
for (<init> ; <test> ; <modify>) {
  <body>
}
```

The `<init>` part is a statement typically of the form

```
int <var>=<val>
```

where the `<var>` part is a variable name that will be used exclusively in the loop, and where `<val>` is its initial value – this is where the term `<init>` comes from.

The `<test>` part is a boolean expression that determines whether or not the body of the loop will be executed.

The `<modify>` part is a statement that normally modifies the `<var>` so that the loop will eventually terminate. A typical `<modify>` statement is of the form `<var>++`.

A `for` loop described above is *exactly* equivalent to the following code using a `while` loop:

```
<init>
while (<test>) {
  <body>
  <modify>
}
```

For example, the following `for` loop

```
for (int i=0 ; i<10 ; i++) {
  System.out.println(i);
}
```

is exactly equivalent to the code

```
int i=0;
while (i<10) {
  System.out.println(i);
  i++;
}
```

This code, when run, should produce the output

```
0
1
2
3
4
5
6
7
8
9
```

Implement the above code

Create a Java program `W1.java` that has a `public static void main` method containing the code

```
int i=0;
while (i<10) {
  System.out.println(i);
  i++;
}
```

as shown above. Run this program and verify that the output is as described above.

Checkpoint 1: Show us your work.

Now create a Java program `F1.java` that is similar to your `W1.java` except that it uses a `for` loop instead of a `while` loop as shown above. Run this program and verify that the output is the same as before.

Checkpoint 2: Show us your work.

What does this code do?

Consider the following method in a file named `W2.java`:

```
public static void main(String [] args) {
    int i=1;
    while (i<10) {
        System.out.println(i);
        i = i + 2;
    }
}
```

Predict what this code does. Draw a state diagram that shows the value of the variable `i` as the program executes. Implement the `W2.java` program and compile and run it.

Checkpoint 3: Show us your state diagram and that your prediction matches your program output.

Convert to a for loop

Write a program `F2.java` that uses a `for` loop instead of a `while` loop and that is equivalent to the program you just tested. Compile and run this program and verify that the outputs are the same.

Checkpoint 4: Show us your program.

Counting down

Write a `while` loop that produces the following output:

```
9
7
5
3
1
```

Think about what your initialization should do, what your test condition should be, and what your modify statement should do.

Implement this code in a program named `W3.java` – as before, the code should go in your `main` method.

Checkpoint 5: Show us your program and that your output matches what is shown above.

Now convert your `while` loop into a corresponding `for` loop in a program named `F3.java`. Run this program and note the output.

Checkpoint 6: Show us your program and output.

Using ArrayLists

Create a FANG "game" called `Quirkles`, in a file named `Quirkles.java`. This game should have a field named `circles` which is of type `ArrayList<OvalSprite>`. In your `setup` method, create between 50 and 100 circles (chosen randomly) each with diameter randomly chosen between 0.05 and 0.1 with a random color, and located randomly on the game canvas. Use a single `OvalSprite` variable named `c` in this code to create each circle, and add the circle both to the game canvas and to the `circles` array list. You will need to "new up" the `circles` variable at the beginning of your

`setup` code. Use a `for` loop to create your circles, add them to the game canvas, and add them to your array list.

Compile and run this program to verify that the circles are appearing appropriately

Checkpoint 7: Show us your work at this point.

Now implement the `advance` method in your `Quirkles` game. In this method, you should randomly change the color of each of the circles you have saved in the `circles` array list. First, get the size of the list using the `size` method, and then march through the circles in a `for` loop, using the `get` method to get a reference to the circle and using the `setColor` method to change its color to a random color.

Compile and run this program.

Checkpoint 8: Show us your work at this point.

Making a Circle class

Create a class `CircleSprite` that extends `OvalSprite`. You will need to edit a new file to do this. The constructor for this class should have one parameter, a `double` that is the diameter of the circle. This parameter should be used with the `super` method to create the appropriately sized `OvalSprite`. The other sprite methods such as `setLocation`, `setColor`, and so forth will be inherited from the `OvalSprite` class.

In your `Quirkles.java` program, replace all of the instances of `OvalSprite` in your program to `CircleSprite` instead. The only other change you should need to make in your `Quirkles` program is to modify how the `CircleSprite` constructor is used to "new up" a `CircleSprite`, since a `CircleSprite` takes only one parameter, not two.

Compile and run this program.

Checkpoint 9: Show us your work at this point.

Changing the circle colors

Now change your `CircleSprite.java` program so that it overloads the `setColor` method. This `setColor` method will have no parameters, but it will set the color of the sprite to a random color. This is an easy two-liner. First, get a referece to the current game using the `getCurrentGame()` method in the `Game` class, and then use this to get a random color using the `randomColor()` method applied to the current game object. This random color should be passed to `super` to set the oval sprite's color.

Finally, in your `advance` method in your `Quirkles.java` program, instead of calling `setColor` with a random color argument, call `setColor` with *no* arguments.

Compile and run this program.

Checkpoint 10: Show us your work at this point. Be prepared to explain how calling `setColor` with no arguments works.