Loops and Conditions

7/5 CIS 110 Recitation

Review: Variable Scope

The scope of a variable is the part of the program in which the variable can be accessed.

It is generally a good idea to minimize the scope of a variable. This reduces the possibility of accidental modification and variable naming conflicts.

Why won't the following lines of code compile?

```
if (10 > 9) {
    String answer = "correct";
}
System.out.println(answer);
```

Review: for-loop

```
for (initialization; condition; update) {
    // statements
}
```

Initialization happens once before the loop starts.

Condition is checked before each iteration.

Update is executed after each iteration.

Note: the compiler will not check whether the initialization, condition, and update expressions are related.

How many times will each loop run?

1. `for (int i = 1; i <= 10; i++)`

2. `for (int i = 10; i > 0; i--)`

3. `for (int j = 1; j <= 10; j = j * 2)`

4. `for (int k = 1; k <= 10 && false ; k++)`

Code Exercise: StringReverser

Write a program that will take in a String as a command-line argument and print out the String but in reverse order. For example, java StringReverser “**CIS110**” will print out: **011 SIC**.

For this exercise, you will need to use the charAt(int) method. The charAt method takes a String and integer (k) as inputs and return the char at the (k+1)th location. (Remember that in CS, we count from 0!). For example:

```
String x = “example”;
char thirdCharacter = x.charAt(2);
System.out.println(thirdCharacter); // prints out the character a
```

Code Exercise 2: StringReverser

If you used a for-loop, try writing the same thing in a while-loop.

If you used a while-loop, try writing the same thing in a for-loop.