CMPSC 111
Introduction to Computer Science I
Fall 2013
Bob Roos
http://cs.allegheny.edu/~rroos/cs111f2013

Recitation 8
17–18 October 2013
Due in Sakai by midnight
“Checkmark” grade

Summary

More about loops.

Last week you learned how to write loops that “count” from a starting value to an ending value. This week you’ll learn about a more general kind of loop, the `while`-loop.

An Example

Suppose you want to play a guessing game with the user of the program. The user must try to guess a number between 1 and 100. If the user’s guess is wrong, print out a helpful hint, such as “that’s too high” or “that’s too low”. Keep repeating this until the user guesses correctly.

The file “`Guess.java`” (on the course web site) contains a program to do this. Here’s the critical section; recall that the symbol “`!=`” means “not equal to”:

```
 ...
int numTries = 0;   // number of guesses made by user
int userGuess = -1; // initialize guess to a wrong answer
while (userGuess != answer) // repeat while user has not guessed correctly
{
  System.out.print("Guess a number between 1 and 100: ");
  userGuess = scan.nextInt(); // read user's response

  if (userGuess == answer)  // First possibility: User got it!
  { System.out.println("Correct!"); }

  else if (userGuess < answer) // Second possibility: guess too low
  { System.out.println("Too low!"); }

  else    // Third (and only remaining) possibility: guess too high
  { System.out.println("Too high!"); }

  numTries ++; // add one to the number of tries
}
```

```
  ...
```

Download it and try it; then study it carefully.

The `while`-Loop

The general form of a `while`-loop is:

```
while ( condition)
{
        ... one or more Java statements ...
}
```

As long as the *condition* is true, the loop will keep repeating. The condition is tested at the "top" of the loop. (This is why we had to initialize the "`userGuess`" to an incorrect value—we want to make sure the loop executes at least once.)

Here's another example: keep generating random throws of a pair of dice until a "double" (both dice the same) is rolled (this is also on the Web site):

```
  ...
int d1 = -1, d2 = -2; // two dice (initialized to different values)
int numTries = 0;
while (d1 != d2) // keep generating random rolls until two are equal
{
    d1 = rand.nextInt(6)+1;
    d2 = rand.nextInt(6)+1;
    numTries++;
}
System.out.println("After " + numTries + " rolls of the dice, two were equal");
  ...
```

Got it? Then write a Java program that:

- Lets the user try to guess a randomly-chosen letter of the alphabet, giving the same "higher" or "lower" advice as in the number guessing game above;
- Keeps adding up values entered by the user until the user enters a value of zero, then prints the sum;
- Generates random integers in the range from 1 to 10 until their sum is greater than 50, then prints the sum and the number of values generated;
- One other looping problem of your own devising.

At the end of the period, or by midnight of the day of your recitation, upload the file you just created.

If you were unable to complete the exercise and have nothing to upload, please send me an email with the subject line "Recitation 8" and tell me what problems you encountered so that I can help you. (Actually, email me if you had any problems or questions, even if you uploaded something.)

General Guidelines for Recitation Sessions

- **Experiment!** Recitation sessions are for learning by doing without the pressure of grades or "right/wrong" answers. So try things! The best way to learn is by trying things out.
- **Submit *something*.** Your grade is just 0 or 1, depending on whether or not you attempt the work and upload something to Sakai.
- **Try to Finish During Class.** Recitation exercises are not intended to be the equal of laboratory assignments. If you are simply a slow typist, I've given you until the end of the day, but ideally you should upload a file, even a non-working one, by the end of the class period and be finished with it.
- **Help One Another!** If your neighbor is struggling and you know what to do, offer your help. Don't "do the work" for them, but advise them on what to type or how to handle things.
- **Review the Honor Code policy on the syllabus.** Remember that you may discuss programs with others, but programs that are nearly identical to others will be taken as evidence of violating the Honor Code.