Repetition - For Loops

Mangat

Motivation

- We often build loops like this:

```
int x = 0;
do {
        ….
        x=x+1;
} while(x<10);
```

Motivation

- A loop that repeats a fixed number of times is so common that there exists another loop specifically for the task
- The FOR Loop avoids having to create a new variable just for counting loops
- It also removes the need for explicitly putting a counter statement directly in the loop

For Statments

```
for (int count = 0; count < 10; count++) {
    System.out.println("This is loop number: ");
    System.out.println(count);
}
```

- A **for statement** is a little more complex than a while/dowhile
- It has a built in **counter** which allows you to decide exactly how many times the program should loop
- You can also use the value of the counter inside of the for loop
- The counter variable does not exist outside of the for statement

For statements

```
for (int count = 0; count < 10; count++) {
    System.out.println ("Hello!");
}
```

- The for statement has three parts
- The first section is for creating a counter. You can name this whatever you want. You also have to decide what value it should be to start.
- The second section is the condition that must be satisfied for the loop to continue
- The third section specifies how the counter should be **incremented**. (count++ is short for count = count + 1)

For statement Examples

```
for (int i = 0; i < 10; i++) { … }

for (int c = 0; count < 10; c = c + 2) { … }

for (int count = 0; count < num; count++) { … }

for (int count = 100; count >= 10; count--) { … }
```

Break and System.exit()

- You can use *break;* to exit a loop manually
- You can use *System.exit(0)* to end you program abruptly

- These should not be used often. Most of the time the code can be re-written to make better use of the loop exit conditions.

- Overuse of these commands can lead to difficult to follow code

Questions?