

CS 152
Computer Programming
Fundamentals
Control Structures: If and While

Brooke Chenoweth

University of New Mexico

Spring 2024

Control Structures

- A sequence of instructions with no choices is not very complicated.
- More complex behaviour is created with *control structures*.
 - Branches – Choose course based on a test
 - Loops – Repeat a sequence of instructions

Branch with if

```
if (booleanExpr) {  
    // statements if booleanExpr is true  
} else {  
    // statements if booleanExpr is false  
}
```

If Example

```
int x = 5;

if (x < 10) {
    System.out.println("x is small");
} else {
    System.out.println("x is big");
}
```

Can omit else

If you only need to choose between doing something or not doing it, you can omit the else part.

```
int x = 5;

if(x == 5) {
    System.out.println("Success!");
}
```

Chaining ifs together

A sequence of multiple ifs lets you chose between several possibilities.

```
if (x < 10) {  
    System.out.println("x is small");  
} else if (x < 20) {  
    System.out.println("x is medium");  
} else {  
    System.out.println("x is big");  
}
```

Nesting ifs

if statements can be used wherever a statement can occur, including inside the body of another if statement.

```
if (x % 2 == 0) {  
    System.out.println("x is even");  
  
    if (x != 2) {  
        System.out.println("x is not 2");  
    }  
}  
else {  
    System.out.println("x is odd");  
}
```

While Loop

```
while (booleanExpr) {  
    // statements to repeat while booleanExpr is true  
}
```

Make sure that your loop condition will be false eventually so that you don't end up with an infinite loop.

Counting with a loop

```
int number = 1;

while (number < 6) {
    System.out.println(number);
    number++; // Remember, ++ adds one to variable
}
```

while and if together

```
int i = 0;

while (i < 10) {
    if (i % 2 == 0) {
        System.out.println(i + " is even");
    } else {
        System.out.println(i + " is odd");
    }
    ++i;
}
```