

# Introduction to Programming

*with Java, for Beginners*

Conditionals(If statements)

## Limitations of sequential programming

- Cannot choose whether or not to perform a command/instruction
- Cannot perform the same command more than once
- Such programs are extremely limited!

ESE112

1

## Control Structures

- Allow a program to base its behavior on certain conditions
- Two kinds:
  - Conditional (If) Statements
  - Loop Structures

ESE112

2

## Recap: Boolean

- **Boolean** is one of the eight primitive types
  - Only 2 values: true, or false
  - Booleans are used to make **yes** or **no** decisions
  - All control structures use Booleans
- The following expression each give a Boolean result:
  - (25 > 24) && (12 == 13) //results to **false**
  - (25 > 24) || (12 == 13) //results to **true**
- Thus based on certain conditions we can alter the outcome or flow of the program

ESE112

3

## Conditionals (“if” statements)

- An “if” statement is a *flow control* statement
- It is also called a *conditional*, or a branch
- We’ll see several “flavors”
  - An “if” all by itself
  - An “if” with an “else” part
  - An “if” with an “else if” part

ESE112

4

## “if” statement

```
if (condition){  
    statement(s)  
}
```

*If the condition is true, then the statement(s) (i.e. instructions) will be executed. Otherwise, it/they won't.*

```
//Assume x is an integer  
if(x > 10) {  
    x = x * 2;  
    System.out.println("x = " + x);  
}
```

ESE112

5

## If statement (contd..)

- { } indicates the block of code that will get executed given the condition is true
- You can avoid the curly brace after condition if only one statement is to be performed
  - If using Dr Java Interaction pane, best to use { }

```
//Assume x is an integer  
if(x > 0)  
    System.out.println(x + " is positive");
```

ESE112

6

## “if-else” statement

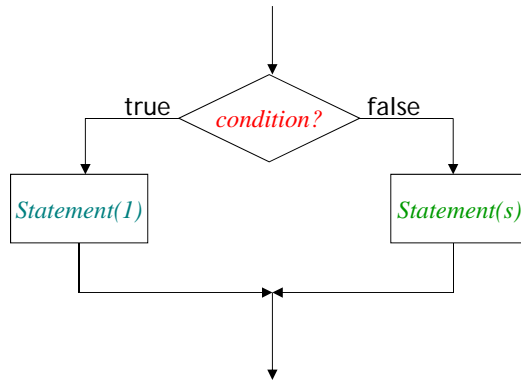
```
if (condition){  
    statement(s)  
}  
else {  
    statement(s)  
}
```

```
//Assume x is an integer  
if(x > 0) {  
    System.out.println(x + " is positive");  
}  
else {  
    System.out.println(x + " is negative");  
}
```

ESE112

7

## If-else Flow chart



ESE112

8

## Style Rule: Indentation and Spacing

- Recommended indentation is from 2 to 4 spaces, but *must be consistent* throughout the program
  - In Dr Java you can set the indent level:  
Edit > Preferences > Miscellaneous
- Single space around every binary operator, including comparisons and assignment (=)

```

if (x < 10) {
    x = x + 1;
}
else {
    x = x - 1;
}
    
```

Indentation points to the space before the opening brace of the first block. Spacing points to the space before the closing brace of the first block.

ESE112

9

## Cascading “if-else”

### Example

```

//Assume variable score is entered by user
if (score > 90)
    System.out.println("Grade A");
else if (score > 80)
    System.out.println("Grade B");
else if (score > 65)
    System.out.println("Grade C");
.
.
else
    System.out.println("F");

//Note: You can avoid the curly brace after
condition if only one statement is to be performed
    
```

ESE112

10

## Nested if-statements

### An if within an if

```

if (condition1){
    if (condition2){
        statement(s) A
    }
    else{
        statement(s) B
    }
}
else {
    statements(s) C
}
    
```

### Truth Table

What values must the conditions have in order for block A to run? B? C?

	A	B	C
condition1	T		
condition2			

ESE112

11

## The infamous "dangling else"

Code	Notes
<pre>if (x &gt; y)   if (y &lt; z)     statementA; else   statementB;</pre>	<p>When is <i>statementB</i> executed?</p> <p>In other words, which <i>if</i> is the <i>else</i> paired with?</p>

- An *else* is paired with the last *else-less if*, regardless of spacing, unless `{ }` dictate otherwise.

```
if (x > y){
  if (y < z){
    statementA;
  }
}
else{
  statementB;
}
```

ESE112

12