Silver Oak Elementary

A National Blue Ribbon School

Primary Programming Essentials Lesson 3

Attila Lengyel

Peter Walker

Victor Vong

Eric O'Riley

Thomas Baker

7/1/2015

Review of Data

- **Data**
  - Numbers, Text, Images, etc.
    - Like the ingredients to a Brownie Recipe
  - Data is stored within variables
  - Variables are like boxes that can only hold one value

Review of Data

- There are special rules for naming variables…
- Are these names possible?
  - 12345_my_number ✗
  - ++hello ✗
  - Zeta_prime ✓
  - My_name_iS_PETer ✓

Review of Data

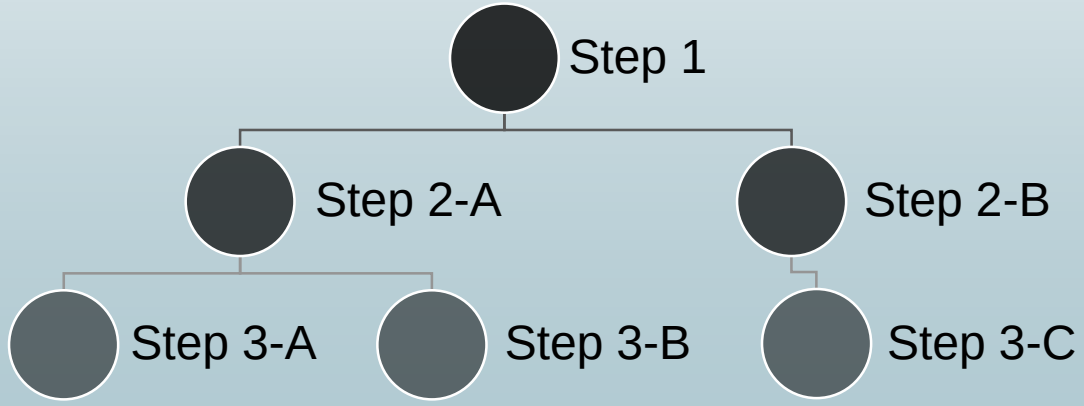
- What is the code to define an **int** variable named **theAnswer** with the value **42**?

```
int theAnswer = 42;
```

Control Flow Statements

- So far, everything has been running sequentially

- But what if we want to have the program do two different things, or three different things, or more?



Control Flow Statements

The *if-then-else* Statement

- The if-then-else statements are our forks in the road, where we can decide to do one bit of code, or another.

- Only *IF* a certain condition is true do we *THEN* execute a block code that has been sectioned off.
- *ELSE*, we execute a separate block of code.
  - Having an else block is optional. It can be left empty

Control Flow Statements

The *if-then-else* Statement

- The if-then-else statement can be thought of more simply as…

IF a condition is **true**…

…then do this.

ELSE…

…do that.

Control Flow Statements

The *if-then-else* Statement Syntax Rules

- The IF always comes first
- The condition is enclosed in parentheses → **(** and **)**
- All code to be executed if the condition is **true** must be enclosed within curly brackets → **{** and **}**

```
if (condition) {
    /* some code to run */
}
else {
    /* some other code to run */
}
```

Logical Operators

Equality and Relations Operators

==	equal to
!=	not equal to
>	greater than
>=	greater than or equal to
<	less than
<=	less than or equal to

Logical Operators

Will the block be executed?

```
if (3 == 3)
{
    Execute Block 1!
}
else
{
    Execute Block 2!
}
```

This one will execute!

- **==** *equal to*
- **!=** *not equal to*
- **>** *greater than*
- **>=** *greater than or equal to*
- **<** *less than*
- **<=** *less than or equal to*

Logical Operators

Will the block be executed?

```
if (3 != 3)
{
    Execute Block 1!
}
else
{
    Execute Block 2!
}
```

This one will execute!

- **==** *equal to*
- **!=** *not equal to*
- **>** *greater than*
- **>=** *greater than or equal to*
- **<** *less than*
- **<=** *less than or equal to*

Logical Operators

Will the block be executed?

```
if (3 < 3)
{
    Execute Block 1!
}
else
{
    Execute Block 2!
}
```

This one will execute!

- **==** *equal to*
- **!=** *not equal to*
- **>** *greater than*
- **>=** *greater than or equal to*
- **<** *less than*
- **<=** *less than or equal to*

Logical Operators

Will the block be executed?

```
if (3 <= 3)
{
    Execute Block 1!
}
else
{
    Execute Block 2!
}
```

This one will execute!

- **==** *equal to*
- **!=** *not equal to*
- **>** *greater than*
- **>=** *greater than or equal to*
- **<** *less than*
- **<=** *less than or equal to*

Logical Operators

Will the block be executed?

```
if (3 > 3)
{
    Execute Block 1!
}
else
{
    Execute Block 2!
}
```

This one will execute!

- **==** *equal to*
- **!=** *not equal to*
- **>** *greater than*
- **>=** *greater than or equal to*
- **<** *less than*
- **<=** *less than or equal to*

Control Flow

Programming Activity!!

If you finish quickly, ask for
the more challenging assignment..

Control Flow

END