EDUCATIONAL OUTREACH
FLORIDA POLYTECHNIC
UNIVERSITY

FLORIDA POLYTECHNIC
GUIDED INTRODUCTION TO PROGRAMMING
Logic: If and Else Statements

Overview:

Lessons 1 and 2 have covered the simple math and the print and scan functions we will be using in the future. Next we will move into logic. The best place to start learning about logic is with the *if* and *else* statements. On top of learning these statements, students will also perform a few simple logic operations which include: is equal, not equal, less/greater than, less/greater than or equal to, “or” statements, and “and” statements. After this lesson you will have a basic knowledge of how logic in a computer works, as well as being able to do simple logic operations to evaluate inputs.

Learning Outcomes

-Students will explain logic in terms of programming and code

-Students will interpret logic operations to create *if* and *else* statements

- Students will create If and Else statements in C to demonstrate what they learned about logic.

Materials

- Computer with internet connection
- Website for the online compiler https://replit.com/languages/c
- Vocabulary list (optional)

The Activity

Part 1

First, start by reviewing and understanding these new logic operations as they will be important to creating the logical *if* and *else* statements. The chart below lists logic operations, how they are used and what they do. This information will be important for this lesson.

Logic Operations	How to use it	What it does
Equals ==	X==Y	Instead of making X equal to Y this arithmetic detects if X is equal to Y

Not Equal !=	X!=Y	Detects if X is not equal to Y
Less Than <	X<Y	Detects if X is less than Y
Greater Than >	X>Y	Detects if X is greater than Y
Less Than or Equal to <=	X<=Y	Detects if X is less than or equal to Y
Greater Than or Equal to >=	X>=Y	Detects if X is greater than or equal to Y
Modulus %	X%Y	Modulus is a math operation that is usually used in logic. It divides X by Y and gives you a remainder

Next, to understand the “or” and “and” statements we need to discuss what logic in computing is. Logic in computing is in a simple sense a series of true or false statements. Using the arithmetic above we can make true and false statements with what we have seen. For example, X==Y, if X is equal to Y then the statement returns true, and if X is not equal to Y then the statements returns false. The truth and falsity of a statement determines what the computer does next in the programs you write. This will be shown when we get into the *if* and *else* statements.

The arithmetic example above is a simplified version of how logic works in computing. In a more mechanical sense logic still does work on a true/false system in computers but we know this as binary. Binary is the lowest level language that a computer uses and is what we call machine code. This is because a computer is a series of logic processes that result in a 1 or a 0, where 1 is true and 0 is false. It is important to know this for the future if you plan to do anything with computers.

Part 2

Once you can explain logic and binary, the next topic to discuss is the "or" and "and" statements. The "or" and "and" statements take the input of two of the logic operations in the chart above in Part 1. The statements can be two different operations or the same with just different parameters. In simple terms, the "or" statement will take both logic operations and uses them to determine its output. We see this in a true false table to visualize the output of the statement.

Logic Operation 1	Logic Operation 2	Result
True	True	True
True	False	True
False	True	True
False	False	False

Similar to the "or" statement, the "and" statement also compares two logic operations to return either true or false. However, in the "and" statement both operations must be true to return true, otherwise the whole statement is false. See the true false table below:

Logic Operation 1	Logic Operation 2	Result
True	True	True
True	False	False
False	True	False
False	False	False

If you are having a hard time understanding this, do not worry. Logic may be a new concept to many of you and that is okay. We all need to start somewhere, and we cannot be expected to understand every new concept as we are given it. There are more examples at the end to help you better understand how this works. Next, are *if* and *else* statements.

Part 3

The *if* and *else* statements are what we refer to in computing as conditional statements, meaning that a condition needs to be met to in order to do what is in the statement. For example, let's say you want to buy a new video game that cost $60, and you only have $50 in your bank account, we can write a simple *if* and *else* statement using this, and it would look like this:

If Bank Account is equal to $60 then buy the game, else save money.

The *if* statement takes a condition, you give it and if that condition is met then it will do what is in the statement, and if the condition is not met it either stops or can go to an *else* statement where it will run the function in the *else* statement. Now let's see what this looks like in code:

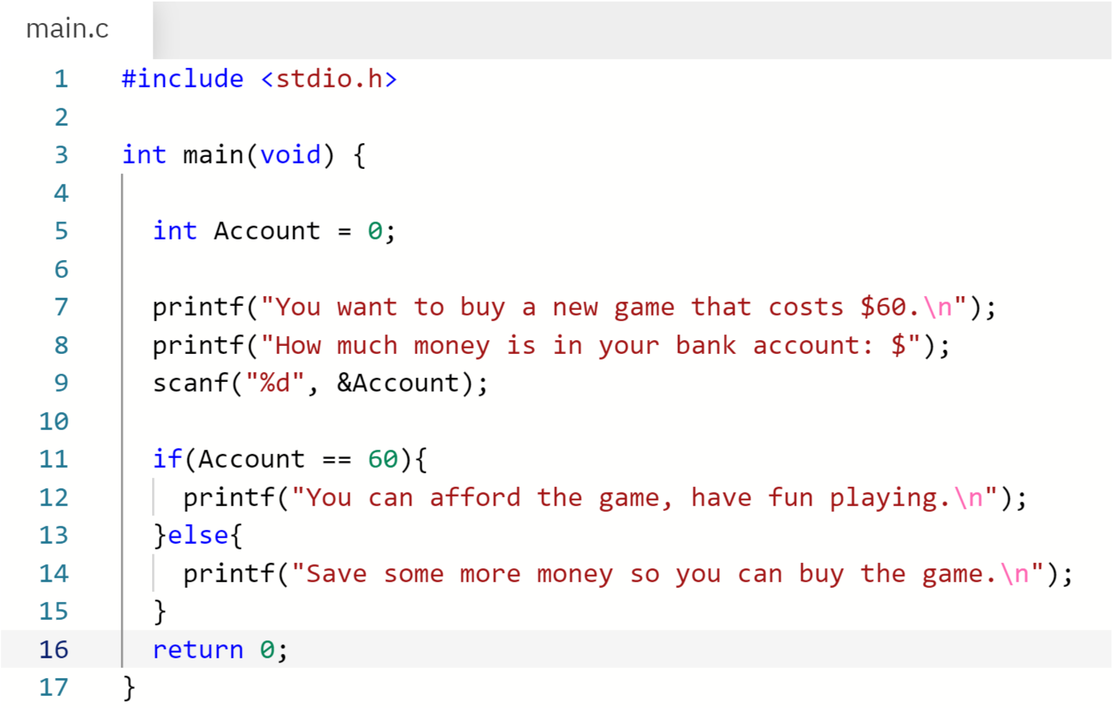
If(condition){

Function;

}else{

Function;}

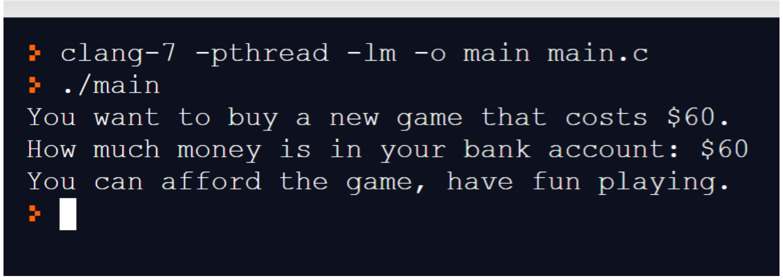
This is the proper way to write an *if* and *else* statement in *C*. So, let me show you the example I made above in code so you may better understand the concept.



main.c

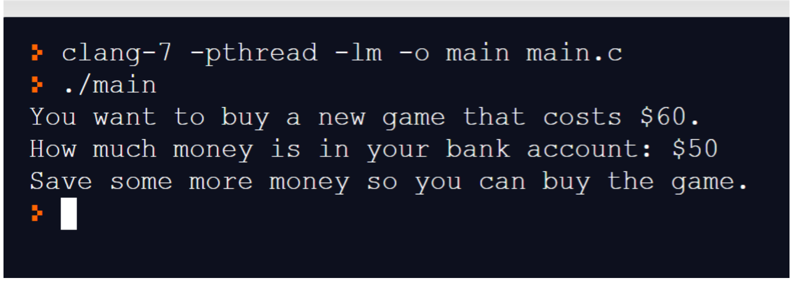
```c
#include <stdio.h>

int main(void) {

    int Account = 0;

    printf("You want to buy a new game that costs $60.\n");
    printf("How much money is in your bank account: $");
    scanf("%d", &Account);

    if(Account == 60){
        printf("You can afford the game, have fun playing.\n");
    }else{
        printf("Save some more money so you can buy the game.\n");
    }
    return 0;
}
```

This is the output you get when you have $60



```
> clang-7 -pthread -lm -o main main.c
> ./main
You want to buy a new game that costs $60.
How much money is in your bank account: $60
You can afford the game, have fun playing.
> █
```

And this is the output you get when you don't have $60



```
> clang-7 -pthread -lm -o main main.c
> ./main
You want to buy a new game that costs $60.
How much money is in your bank account: $50
Save some more money so you can buy the game.
> █
```

It is also important to note that you can write just an if statement and not have the else after it. Knowing how to use the *if* and *else* statements is essential in programming. This is because they are used to do almost any high-level program, but for now it is good that you just get some simple experience with it.

Assessment

1. With the following information, write a conditional statement:
You want to drive to Disney, and you are 50 miles away, but you only have 20 miles worth of gas, can you make it without stopping?

2. Reading the definition of the modulus function, what is 15%4?

3. What is 15%5?

Extensions and/or Additional Resources

- Link to Lesson 3 Video
- Link to Programming Vocabulary
- https://www.w3schools.com/

- https://www.geeksforgeeks.org/c-programming-language/
- https://replit.com/languages/c

Created By:
Ryan Floyd, Computer Science, '21

Thank you for downloading this lesson, please take a moment to complete our survey