Lab 5

If Statements

Learning Objectives

Until now you have been writing linear programs that are executed in a straight line, one statement after another. After completing this lab you will be able to write branching statements that cause one section of code to be executed or not executed, depending on a conditional.

Basic Lab Instructions!

- Talk to your classmates for help.
- You may want to bring your textbook to future labs to look up syntax and examples.
- Stuck? Confused? Have a question? Ask a TA/Lab Engineer for help, or look at the book or past lecture slides.
- Complete as many problems as you can within the allotted time. You don't need to keep working on these exercises after you leave the lab.
- Before you leave today, make sure to check in with one of the Lab Engineers/TAs in the lab to get credit for your work.

Lab Tasks

Creating a New Project

After completing this section you will have created a project for today's lab.

1. Open the solution CS110Labs you created in the lab last week.
2. Add a new project called Lab04 within the solution CS110Labs. Set the project as startup project (right-click the project in Solution Explorer). STOP HERE! start with the lab tasks, and refer to the following steps accordingly.
3. Click File in the top menu bar of Visual Studio, and then select the Add New Item option (Add Existing Item option depending on the lab task).
4. Select C++ File
5. Type in the code listed.
6. Compile (Build Lab04) the program and execute it (Execute without Debugging).

Task #1: What If

What if in a post-apocalyptic world overrun by zombies has only one chance at recovery—a team of robots that must protect and clone a lone surviving human baby. They just need to do this amidst the endless hordes of zombies who have been driven rabid by their frustration at inedible bots and their lust to eat the one living brain left on the planet. (Zombies vs. Robots)

The following program introduces you to the if statement. Type in the code, save as WhatIf.c, and make it run.

```c
#include <stdio.h>
#include <stdlib.h>

int main()
{
  int zombies = 20;
  int robots  = 30;
  int baby    = 1;

  if ( robots < zombies )
  {
    printf( "Too many zombies! The baby is doomed!\n" );
  }

  if ( robots > zombies )
  {
    printf( "Not many zombies! The baby is saved!\n" );
  }

  if ( baby == 0 )
  {
    printf( "The human race is extinct!\n" );
    printf( "The zombies starve to death!\n" );
  }

  if ( baby != 0 )
  {
    printf( "The zombies are crazy for some brains!\n" );
  }

  return EXIT_SUCCESS;
}
```

What You Should See

```
Not many zombies! The baby is saved!
The zombies are crazy for some brains!
```

What You Should Do on Your Own

Guess what do you think; respond by a comment at the end of the source file.

1. What do you think the if does to the code under it?
2. What is the purpose of the curly braces in the if statement?
3. Change the values of the variables so that a different set of messages are printed.

You have to turn-in the modified WhatIf.c.

Task #2: How Old Are You?

Write a program HowOldAreYou.c which displays a different message depending on the age given. Here are the possible responses:

- ❖ age is less than 16, say “You can’t vote."
- ❖ age is less than 18, say “You can’t drive."
- ❖ age is less than 25, say “You can’t contest for election."
- ❖ age is 25 or over, say “You can do anything that’s legal."

A sample run of your program.

```
Hey, what's your name? Hamza

Ok, Hamza, how old are you? 17

You can't drive, Hamza.
You can't contest for election.
```

You have to turn-in HowOldAreYou.c.

Task #3: Else And If

Pure water is tasteless, odorless, and colorless. Water can occur in three states: solid (ice), liquid, or gas (vapor). Open ElseAndIf.c in your project, and make it run.

```c
#include <stdio.h>
#include <stdlib.h>

int main()
{
  int temp = 30;

  if( temp < 32 )
  {
    printf( "Water is in solid ice state.\n" );
  }
  else if( temp < 212)
  {
    printf( "Water is in liquid water state.\n" );
  }
  else
  {
    printf( "Water is in gaseous steam state.\n" );
  }

  return EXIT_SUCCESS;
}
```

What You Should See

```
Water is in solid ice state.
```

What You Should Do on Your Own

Guess what do you think; respond by a comment at the end of the source file.

1. What do you think else if and else are doing?
2. Remove the else part at the beginning of the else if statement. What difference does that make? Why?

You have to turn-in the modified ElseAndIf.c.

Task #4: Weekday Name

Open the program WeekdayName.c in your project that returns the name of a day of the week given the day number. Compile, and execute it. You should notice that the code is incomplete, use if and else to complete it according to the following table:

Number	Day of week
1	Saturday
2	Sunday
3	Monday
4	Tuesday
5	Wednesday
6	Thursday
7	Friday
anything else	error

What You Should See

```
Weekday 1: Saturday
Weekday 2: Sunday
Weekday 3: Monday
Weekday 4: Tuesday
Weekday 5: Wednesday
Weekday 6: Thursday
Weekday 7: Friday

Weekday 17: error
Weekday -1: error
```

You have to turn-in the modified WeekdayName.c.

Task #5: How Old Are You, Specifically?

Using if statements, else if, and else statements, make a program HowOldAreYou2.c which displays a different message depending on the age given.

age	message
less than 16	"You can't drive."
16 to 17	"You can drive but not vote."
18 to 24	"You can vote but not rent a car."
25 or older	"You can do pretty much anything."

Sample output.

```
Hey, what's your name? (Sorry, I keep forgetting.) Hamza
Ok, Hamza, how old are you? 17

You can drive but you can't vote, Hamza.
```

You have to turn in HowOldAreYou2.c.

5.0.1 Task #6: Space Boxing

In the year 3000, all planets in the solar system are terraformed to make it habitable for humans. From birth people are genetically designed to fit one of the five castes, each with their contribution to the society. You a beta are trained to become an interplanetary scout. Occasionally you have to take a break on one of the colonized planets. However, you have to be careful about the effect of surface gravity on different planets—lower or higher—either hit something after taking springy steps or pull a muscle under stress respectively. A program to keep track of weight changes should come in handy.

Write a program PlanetaryWeight.c that should ask your earth weight, and a number for the planet visiting. It should then compute your weight on the destination planet based on the table below:

#	Planet	Relative gravity
1	Mercury	0.378
2	Venus	0.907
3	Earth	1
4	Moon	0.165
5	Mars	0.377
6	Jupiter	2.364
7	Saturn	0.910
8	Uranus	0.889
9	Neptune	1.125

For example, if you weighs 75 kgs. on earth, then he would weigh just under 30 kgs. on Mars, since Mars' gravity is 0.377 times earth's gravity. (75 is 28.275)

Sample output.

```
Please enter your current earth weight: 75

I have information for the following planets:
   1. Mercury 2. Venus   3. Earth
   4. Moon    5. Mars    6. Jupiter
   7. Saturn  8. Uranus  9. Neptune

Which planet are you visiting? 2

Your weight would be 28.275 kilograms on that planet.
```

You have to turn in PlanetaryWeight.c.

Task 7: A Little Quiz

Write an interactive quiz LittleQuiz.c. It should ask the user three multiple-choice or true/false questions about something. It must keep track of how many they get wrong, and print out a "score" at the end.

Sample output.

```
Are you ready for a quiz (Y/N)? Y
Okay, here it comes!

Q1. What is the capital of Pakistan?
  1. Lahore
  2. Karachi
  3. Islamabad

> 3

That's right!
```

```
12
13 Q2) Can you store the value 3.147 in a variable of type int?
14   1) yes
15   2) no
16
17 > 1
18
19 Sorry, 3.147 is a floating-point. type int can only store numbers.
20
21 Q3) What is the result of 2+6/3?
22   1) 11
23   2) 4
24   3) 8/3
25
26 > 2
27
28 That's correct!
29
30 Overall, you got 2 out of 3 correct.
31 Thanks for playing!
```

You have to turn-in LittleQuiz.c.

Task #8: BMI Categories

The body mass index (BMI) is commonly used by health and nutrition professionals to estimate human body fat in populations. It is computed by taking the individual's weight (mass) in kilograms and dividing it by the square of their height in meters.

Open BMICalc.c you worked on in the previous lab, and modify it using if statements to show the category for a given BMI.

BMI	category
less than 15.0	very severely underweight
15.0 to 16.0	severely underweight
16.1 to 18.4	underweight
18.5 to 24.9	normal weight
25.0 to 29.9	overweight
30.0 to 34.9	moderately obese
35.0 to 39.9	severely obese
40.0 and up very	severely (or "morbidly") obese

Sample Output.

```
1 Your height in m: 1.8
2 Your weight in kg: 73
3
4 Your BMI is 23.83673
5 BMI Category: normal weight
```

You have to turn-in BMICalc.c.

Hand in

Hand in the source code from this lab at the appropriate location on the blackboard system at LMS. You should hand in a single compressed/archived file named Lab_5_<your reg. No. XXX without angle brackets>.zip that contains the following.

1. All completed C source files representing the work accomplished for this lab: WhatIf.c; HowOldAreYou.c; ElseAndIf.c; WeekdayName.c; HowOldAreYou2.c;

PlanetaryWeight.c; LittleQuiz.c; and BMICalc.c. The files should contain author in the comments at the top.

2. An plain text file named OUTPUT.txt that includes a) author information at the beginning, b) a brief explanation of the lab, and c) any comments, or suggestions.

To Receive Credit

1. By showing up on time for lab, working on the lab solution, and staying to the end of the class period, only then you can receive full credit for the lab assignment.

2. Comment your program heavily. Intelligent comments and a clean, readable formatting of your code account for 20% of your grade.

3. In-class lab time is not intended as free time for working on your program assignments. Only if you have completely solved the lab assignment, including all challenges, and have had your work checked off for completeness by your TA/Lab Engineer should you begin the program assignment.