

Keyboard Input

Learning Objectives

After completing this lab you will be able to get input from keyboard, and make your program work on it in C.

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 Lab03 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 Lab03) the program and execute it (Execute without Debugging).

Task #1: Asking Questions

You have been doing a lot of printing, what we want to do now is to get data into your programs. The way to do this might seem strange today, but after a couple of labs it will make sense.

You all agree what most of the software does is,

1. take some input from a user.
2. change it.
3. print out to show how it changed.

You have done the last two steps many times, next exercise will explain how to take input from a user. If you have no idea what “input” means, completing the next exercise will help you understand it.

```

1  #include <stdlib.h>
2  #include <stdio.h>
3
4  #define MAX_STRING_LENGTH 80
5
6  int main()
7  {
8      int age;
9      char height[MAX_STRING_LENGTH];
10     float weight;
11
12     printf( "How old are you? ");
13     scanf( "%d", &age );
14
15     printf( "How tall are you? ");
16     scanf( "%s", &height );
17
18     printf( "How much do you weigh? ");
19     scanf( "%f", &weight );
20
21     printf( "So you're %d old, %s tall and %.1f heavy.\n", age, height, weight );
22
23     return EXIT_SUCCESS;
24 }
```

What You Should See

```

1  How old are you? 30
2  How tall are you? 5'11"
3  How much do you weigh? 167
4  So, you're 30 old, 5'11" tall and 167 heavy.
```

What You Should Do on Your Own

Assignments turned in without these things will not receive any points.

1. Change the program so that it reads in the height in two parts. The first part should read in an int for the number of feet. Then read in a second int for the number of inches. Try to make the output look the same, though.

```

1 How old are you? 30
2 How many feet tall are you? 5
3 And how many inches? 11
4 How much do you weigh? 167
5 So, you're 30 old, 5'11" tall and 167 heavy.

```

You have to turn-in the modified AskingQuestions.c file.

Task #2: Name, Age, and Salary

Write a program NameAgeAndSalary.c that asks the user for their name. Then display their name to prove that you can recall it. Ask them for their age. Then display that. Finally, ask them for how much they make and display that. You should use the most appropriate data type for each variable.

```

1 Hello. What is your name?
2 Hamza
3
4 Hi, Hamza! How old are you?
5 20
6
7 How much do you make, Hamza?
8 300.50
9
10 300.50! I hope that's per hour and not per year!

```

You have to turn-in the NameAgeAndSalary.c file.

Task #3: More User Input of Data

Write a program MoreUserInputOfData.c that asks the user for several pieces of information, and display them on the screen afterward as a summary.

- ❖ first name
- ❖ last name
- ❖ grade (classification)
- ❖ student id number
- ❖ login name
- ❖ GPA (0.0 to 4.0)

You must use the most appropriate type for each variable and not just strings for everything.

```

1
2 First name: Hamza
3 Last name: Mustafa
4 Semester: 4
5 Student ID: 111
6 Login: hamza_111
7 GPA (0.0-4.0): 4.0
8
9 Your information:
10 Login: hamza_111
11 ID: 111
12 Name: Mustafa, Hamza
13 GPA: 4.0
14 Semester: 4

```

You have to turn-in the MoreUserInputOfData.c file.

Task #4: Age in Five Years

Write a program AgeIn5.c that asks the user for their name. Then display their name to prove that you can recall it. Ask them for their age. Then display what their age would be five years from now. Then display what their age would be five years ago.

```
1
2 Hi, Hamza! How old are you? 20
3
4 In five years you will be 25 years old.
5 And five years ago you were 15!
```

You have to turn-in the AgeIn5.c file.

Task #5: A Dumb Calculator

Make a simple numeric calculator DumbCalculator.c. It should prompt the user for three numbers. Then add the numbers together and divide by 2. Display the result. Your program must support numbers with decimals and not just integers.

```
1 What is your second number? 2.2
2 What is your third number? 5.5
3
4 ( 1.1 + 2.2 + 5.5 ) / 2 is ... 4.4
```

You have to turn-in the DumbCalculator.c file.

Task #7: BMI Calculator

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.

Write a program BMICalc.c with a sample output.

```
1 What is your second number? 2.2
2 What is your third number? 5.5
3
4 ( 1.1 + 2.2 + 5.5 ) / 2 is ... 4.4
```

You have to turn-in the BMICalc.c file.

Bonus: Imperial Measurements

For a bonus point, input their weight and height using pounds and inches, and convert to kilograms and meters to figure the BMI.

```
1 Your weight in pounds: 167
2
3 Your BMI is 23.3
```

For an extra bonus point, input their height in feet and inches.

```
1 Your height (inches): 11
2 Your weight in pounds: 167
3
4 Your BMI is 23.3
```

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_4_<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: AskingQuestions.c; NameAgeAndSalary.c; MoreUserInputOfData.c; AgeIn5.c; DumbCalculator.c; and BMICalc.c. The files should contain author in the comments at the top.
2. A 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.