

CS 36 Lab instructions ([Video](#))

1. Upload your file only to the link that is provided for each lab.
2. Copy your lab programs with codes followed by their output to a text editor [notepad++](#) for Windows or for the case of Mac you can use the [TextEdit](#) that comes with Macs. Starting, from question number 1 to the last question for the lab. You must number each lab question and each lab question that is coded must have the number of test runs specified (for example 5 test runs). A test run is defined as executing the program by clicking the compile and run. Failure to follow these instructions will result in a zero for that question.
3. Name your lab file with lastname_firstinitial_labnumber.txt and only upload this single file with all the programs for that specific lab. This .txt file will contain all the programs for your lab. You must name your file exactly as stated. You will get a zero if you do not do so. Example your name is Joe Smith, name your file **lab01_smith_j.txt** for example for lab number 1.
4. Lab must only be uploaded to the link provided. Late lab, and lab file sent to my email or message through Canvas will not be accepted for any reason.

Windows users: use [notepad++](#) , download from the link below

<https://notepad-plus-plus.org/downloads/>

Mac Users : use [TextEdit](#) – comes with your Mac. Use finder to locate the program

Mac Users will have to read information of the link below to work and save file as .txt

<https://beebom.com/how-save-files-txt-format-textedit-mac/>

In the video I have only shown you step 1 to select the plain text button; you must do the rest of the step 2 in beebom.com. Switch over to the **“Open and Save”** tab. We need to change two things here, first, check the checkbox next to **“Add .txt extension to plain text files”**; and second, change the **“Saving files”** encoding to **“UTF-8”**.

Then click format and select Make Plain Text. Start your work. When you save it should prompt you with filename ending with .txt

If you do not set the preferences to save your file as .txt then what you upload to Canvas it is not readable and you will get a zero. So, make sure you go to beebom.com and set that preference before saving your lab file with a .txt extension.

For each question you must have the top banner with your name and lab question number and student ID and brief program description. If the top banner is not provided then that question will be marked as wrong and you will not get any points for that question.

Each lab assignment will have its own specification on the number of test runs required (not necessary 5 as given in this instruction example). Please follow the instructions given in each lab assignment for the required number of test runs.

Continue on next page...

Example of a Lab for submission to Canvas:

Lab1

Question 1

```
/******  
/*Lab 1 Question 1  
/*Name : Sam Smith  
/*Student ID : 123456  
/*Date: 08/19/20  
/*This Program Finds Sum and Average of 3 numbers */  
/*  
/******
```

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

```
int main()  
{  
    //Declaration  
    int a, b, c, sum;  
    float avg;  
  
    //Data/input  
    a = 10;  
    b = 20;  
    c = 30;  
  
    //Processing/Calculations  
    sum = a + b + c;  
    avg = sum / 3;  
  
    //Output  
    printf("The sum for %d %d %d is\n", a ,b ,c);  
    printf("The sum is %d\n", sum);  
    printf("The average is %f\n", avg);  
  
    return 0;  
}
```

```
/*
```

Test run 1

```
The sum for 10 25 30 is  
The sum is 65  
The average is 21.666666
```

Test run 2

```
The sum for 10 20 30 is  
The sum is 60  
The average is 20.000000
```

Test run 3

The sum for 15 25 30 is
The sum is 70
The average is 23.333334

Test run 4

The sum for 0 25 0 is
The sum is 25
The average is 8.333333

Test run 5

The sum for 120 25 5600 is
The sum is 5745
The average is 1915.000000
*/

Lab 1 Question 2

```
/*Lab 1 Question 2
/*Name : Sam Smith
/*Student ID : 123456
/*Date: 08/19/20
/*This Program allows the user to enter 3 integers*/
/*to find the Sum and Average of 3 numbers
/*
/*
*****/

#include<stdio.h>

int main()
{
    //Declaration
    int num1, num2, num3, sum;
    float avg;
    char name[10];

    //Data/input
    printf("What's your name? ");
    gets(name);

    printf("Enter 3 integers ");
    scanf("%d%d%d", &num1, &num2, &num3);

    //Processing/Calculations
    sum = num1 + num2 + num3;
    avg = sum / (float)3;

    //Output
    printf("\nHello, %s\n", name);
    printf("The sum for %d %d %d is\n", num1, num2, num3);
    printf("The sum is %d\n", sum);
    printf("The average is %f\n", avg);

    return 0;
}
```

/*

Test run 1

What's your name? Sam Smith
Enter 3 integers 10 20 30

Hello, Sam Smith
The sum for 10 20 30 is
The sum is 60
The average is 20.000000

Test run 2

What's your name? Sam Sung
Enter 3 integers 10 25 30

Hello, Sam Sung
The sum for 10 25 30 is
The sum is 65
The average is 21.666666

Test run 3

What's your name? Sammy
Enter 3 integers 15 25 30

Hello, Sam Sung
The sum for 15 25 30 is
The sum is 70
The average is 23.333334

Test run 4

What's your name? Johnny Matrix
Enter 3 integers 0 25 0

Hello, Johnny Matrix
The sum for 0 25 0 is
The sum is 25
The average is 8.333333

Test run 5

What's your name? Aeon
Enter 3 integers 120 25 5600

Hello, Aeon
The sum for 120 25 5600 is
The sum is 5745
The average is 1915.000000

*/

Paste the next question here

Lab 1 Question 3

```
/******  
/*Lab 1 Question 3  
/*Name : Sam Smith  
/*Student ID : 123456  
/*Date: 08/19/20  
/*This Program finds the largest of 3 numbers  
/*  
/******
```

Your program codes here

Codes

Codes

. . .

End of program codes

```
/*  
Test runs for Q3  
#Your output for test run 1  
#Some output  
#Some output  
#. . .  
End test run 1
```

And so on . . .

Lab

1. Please first go over the Lab instructions including the video. The instructions show you how to prepare the lab for submission. There is a sample of how a lab submission would look like.
2. You are to put all your programs plus its output(test runs) into one single .txt file.
3. You are to correctly number each of the questions and submit **one file** to the link provided under Lab 1 module.
4. Windows users can download notepad++ and Mac users can use textEdit. Mac users please pay attention on how to set up your textEdit to save as .txt file correctly.

What is a test run?

1. A test run is running a program once. If you are required to provide 5 test runs, then you have to run the program 5 separate times. Sample test runs are usually provided for homework and you must also use the sample test runs provided. So, if a question provided you will one sample test run, then you will also use the data from the sample test run and submit that test run as one of the 5 test runs. Meaning you will provide 4 other test runs yourself to make up the 5 test runs required.

2. The sample test run also shows you the specifications of how the output must look like. Your output must look exactly like the sample test run output given.

for example, the sample test run given:

Enter a number : 20

The number entered is : 20

The number divide by 2 is : 10

Your test run output must look exactly the same as above. You cannot change any of the specifications. The example below changes the sample test run specs and will be graded as incorrect and you will receive a zero for that whole program set. The strings were changed for example, "Enter a number : " was changed to "Hello, please input a number : " and the colon for the output were not aligned. The output for 20 and 10 in the second and third line were not right justified.

Hello, please input a number : 20

Thanks, the number you entered is : 20

Divide the number by 2 is : 10

How do I know if my program runs correctly?

The nice thing about programming is that you know the answer.

If you are to write a program to calculate the area of a rectangle, and the user inputs the length and the width. You can check your answer by using a calculator and multiply the length and the width and then check if the answer from your program matches the calculator. This is call hand tracing.

The sample test run provided for you is a solution to the program you are to code. If your program came up with exactly the same output as the sample test run, then your program should be running correctly. But you should test with other input data to make sure that it will also perform correctly with your own test data.