This is going to be hands on lab where you would be doing everything☺. In this lab, you are given a series of problems which are hopefully going to burn some of your brain cells.

Objectives:

- Learn about variables and use of variables in C++.
- Learn how to declare and initialize a variable.
- Learn about basic data types in C++.
- Learn how to get input from a user in a C++ program.
- Learn how to use basic mathematical operators. (+,-,*,/,%,=).

Variables (Introduction):

As a programmer, you have to decide what types of information are necessary for your program and how this information would be used. When writing a program, you will provide these pieces of information to the computer, the computer then puts them together. When your program runs it is "loaded" into the memory (RAM). When the user is using your program, the information that your application requests also goes into the RAM while your program is processing such requests.
Because your program will be made of many of these things, the computer needs to know what these things would be, and how much space each one of them would need. Because such a thing can change (vary) throughout your program, it is called a variable. By its definition, ***a variable is a place in memory that holds some information***. Some information may be numbers, characters, strings or Booleans.

Declaration:

Every variable must be declared before use or in simple words, Before using a variable, you must first let the compiler know. Letting the compiler know about a variable is referred to "*Declaring*" the variable. The compiler will need two pieces of information concerning each variable: the amount of space the variable will need, and a name to recognize that variable.

Therefore, the formula of declaring a variable is:

Data_Type variable_name;

Rules for variable names:

When using the various necessary variables in your programs, each one of them need to be identified. A variable is primarily recognized by its name. C++ provides rules for naming items in a program.
The name of a variable:

- May start with an underscore "_" or a letter, lowercase or uppercase, such as a letter from a to z or from A to Z. Examples are Name, gender, _Students, pRice
- Can include letters, underscore, or digits. Examples are: keyboard, Master, Junction, Player1, total_grade, _Score_Side1
- Cannot include special characters such as !, %,], or $
- Cannot include an empty space
- Cannot be any of the reserved words.

A name can consist of one word such as country. A name could also be a combination of more than one word, such as first_name or date_of_birth. To separate words in a variable name, programmers usually use the "_" (underscore) sign.
C++ is case-sensitive; this means that AIR, Air, air, and AiR are four completely different words.

Naming Conventions: Keep all the variables in small case.

Data types:

The amount of memory space necessary to store a variable is also referred to as a data type. A data type provides two valuable pieces of information to the compiler: what amount of space the variable will use, what kind of information will be allowed in that space. After declaring a variable, the compiler reserves a space in memory for that variable. It will always be your responsibility to decide how much space a variable needs, based on your goal.

Name	Size (in bits, on x86)
bool	8 (top 7 bits are ignored)
char	8
short	16
int	32
long	32 (can be 64 on other architectures)

If we want to store larger number then we have to use (long long).

Floating point types:

Name	Size (in bits, on x86)
float	32
double	64

C++ cout

The **cout** object is used to display the output to the standard output device. It is defined in the iostream header file.

Here's an example:

```
#include <iostream>

using namespace std;

int main() {
  int a = 24;

  // print variable
  cout << "Value of a is " << a;

  return 0;
}

// Output: Value of a is 24
```

Try the following codes:

1.

```
#include <iostream>
Using namespace std;
void main()
{
   int i = 6;
   int j = 8;
```

```
# include <iostream>
using namespace std;

void main ()
{
        int num1 = 0 , num2 = 0 , sum = 0 ;
        cout << "Enter 2 numbers : " << endl ;
        cin >> num1 >> num2 ;
        sum = num1 + num2 ;

        cout << "sum of 2 numbers is:  " << sum << endl ;

}
```

2

```
#include <iostream>
Using namespace std;

void main()
{
  char first_initial = 'M';
  char middle_initial = 'A';
  char last_initial = 'V';
   cout << “My name initials are: ” <<  first_initial <<  middle_initial << last_initial
<< endl ;
}
```

C++ cin

The cin object is used to accept input from the standard input device i.e. keyboard. It is defined in the iostream header file.

```
#include <iostream>
using namespace std;

int main() {
  int num;

  cout << "Enter a number: ";

  // take integer input
  cin >> num;

  cout << "You entered: " << num;

  return 0;
}
```

Pre/ Post Increment/Decrement Operators:

PRE means do the operation first followed by any assignment operation. POST means do the operation after any assignment operation. Consider the following statements

```
++count;      /* PRE Increment, means add one to count */
count++;      /* POST Increment, means add one to count */
```

Let us examine what happens when we use the operator along with an assignment operation. Consider the following program,

```
# include <iostream>
using namespace std ;

void main()
{
        int count = 0, loop=0;

        loop = ++count;  /* same as count = count + 1; loop = count;  */
        cout << "Loop =  " << loop << "Count =  " << count << endl ;

        loop = count++;  /* same as loop = count;  count = count + 1;  */
        cout << "Loop =  " << loop << "Count =  " << count << endl ;
}
```

Output:

loop = 1, count = 1
loop = 1; count = 2

If the operator precedes (is on the left hand side) of the variable, the operation is performed first, so the statement

```
loop = ++count;
```

really means increment count first, then assign the new value of count to loop.

Try the following piece of code and examine output:

```
# include <iostream>
using namespace std ;

void main()
{
    int x = 5, y = 5;
    cout << "The value of x is  "<< x++;
    cout <<"The value of y is " << ++y;
    cout << "value of x and y " << x << y ;

    x = 10;
    y = ++x;
```

```
        cout <<"The value of y is " << y;

       x = 10;
       y = x++;
        cout <<"The value of y is " <<y;
}
```

Lab Tasks 2:

Problem 1) Calculate the volume of a cone using the formula:
Volume of Cone = $1/3\ \pi\ r^2\ h$
Take the values of 'r' and 'h' from the user. Choose the data types wisely.

Problem 2) Check this mathematical trick for yourself.
Think of a number (and don't forget it). Double it. Add six. Divide your answer by two. Now take away the number you first thought of. The number you get should be... three.

NOTE: Get 1st number from user.

Problem 3) Write a program that takes 5 integers from the user and displays the sum, average and product of these numbers on screen.

Problem 4) Write a program that gets marks of 5 subjects from user and then calculates the percentage and display result on screen.

NOTE: Get marks of Calculus, Programming , E and M , fundamentals of CS, Islamiat from user. Assume total marks of each subject is 100.

Problem 5) Write a program that calculates the remainder and quotient of two numbers entered by user and display the answers.

NOTE: Get numbers from user.

Problem 6) Write a program for the following mathematical trick.
Ask user for any number. Double it. Add 10 to it. Now half the number. Then subtract your first number. The answer will be five.

Problem 7) Write a program to find whole square of three numbers by the following formula

$$(a + b + c)^2 = a^2 + b^2 + c^2 + 2\ (ab + bc + ca)$$

NOTE: Get the values of a, b and c from user.

Problem 8)

Write a program that asks the user for three floating-point values. The program should compute **square of** each value and add them together. Then, display the result.

Problem 9)

Take two inputs from user and write a C++program to perform the following operations

$$x = a^2 + 2ab + b^2$$

Problem 10)

Write a program that calculates the reciprocal of a non-zero integer entered by the user. (The reciprocal of an integer n is 1/n). The program should store the result of the calculation in a variable of type double, and then output it.

Problem 11)

Check this mathematical trick for yourself.

Take user input for 2 single digit inputs, Take first number and double it, add 5 to the result, multiply the result with 5. Add the second number to the answer, subtract the answer with 4, and subtract the answer again with 21. The answer should come out to be the same two single digit numbers.

Note: Perform the whole trick with only one mathematical equation.

Problem 12)

Take user input for a lower case alphabet (small letter) and convert it into upper case (capital letter)

Problem 13)

Take user input for an alphabet and print its ascii value.

Problem 14)

Write C program to take user input for a character from 0 to 9, now print its corresponding decimal (from 0 to 9)

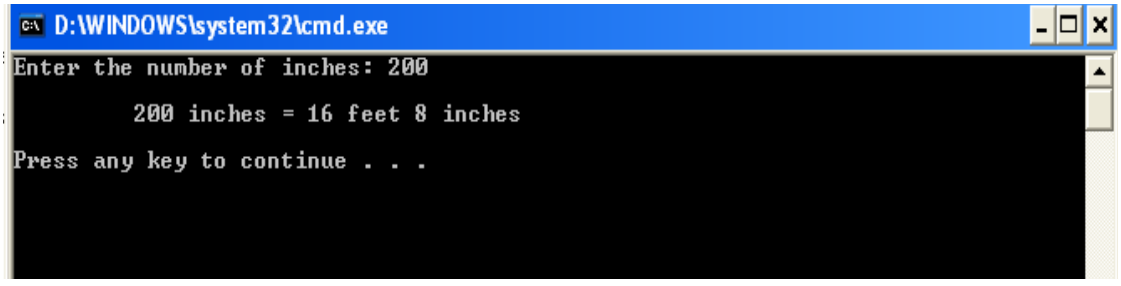
1. using subtraction
2. without using subtraction (type casting)

Problem 15)

Write a program that converts inches to feet-and-inches – for example, an input of 77 inches should produce an output of 6 feet and 5 inches. Prompt the user to enter an integer value i.e. the number of inches, and then make the conversion and output the result.

(1 ft = 12 inches or 1 inch = 0.083333 ft)

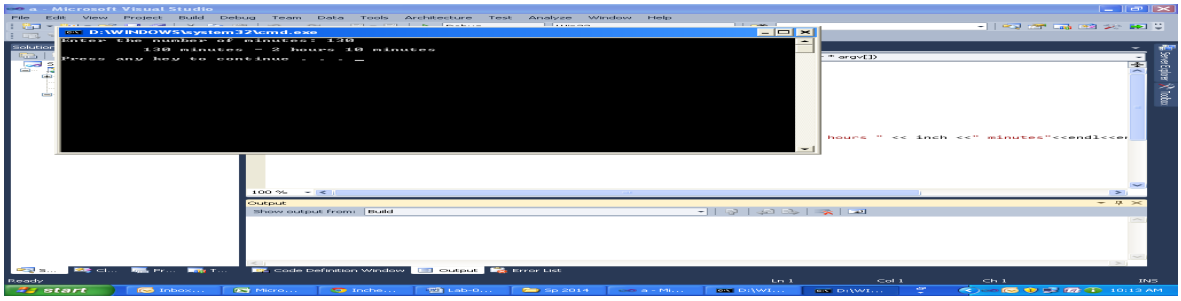
Hint: use **const** to store the inches-to-feet conversion rate, and employ the modulus operator.



Problem 16)

Write a program to get number of minutes from user and convert it into hours and then print the result.

Sample Output:


D:\WINDOWS\system32\cmd.exe
Enter the number of minutes: 130
130 minutes = 2 hours 10 minutes
Press any key to continue . . .

**********GOOD LUCK**********