**Experiment #08**

**Objective:**

**To be familiar with user define functions.**

**To be familiar with user define header files.**

**Theory:**

A function is a self-contained program that carries out some specific, well-defined task. Every C program consists of one or more functions, one of these function must be called main. Execution of the program will always begin by carrying out the instruction in main. Additional functions will be subordinate to main, and perhaps to one another.

A function will carry out its intended action whenever it is accessed (i.e. whenever the function is “called”) from some other portion of the program. The same function can be accessed from several different places within a program. Once the function has carried its intended action, control will be returned to the point from which the function was accessed.

Generally function will process information that is passed to it from the calling portion of the program, and return a single value. Information is passed to the function via a special identifier called arguments (also called parameters). And return via the return statement.

**Defining a function**

To define a function in the program, it requires three type of the information.

**Function declaration.**

Function call.

Function definition

Function declaration:

Just as you can not use a variable without first telling the compiler what it is, you also can not use a function without telling the compiler about it. Functions are declared by specifying the name of the function, no of the arguments and return type of the function.

e.g. return type name\_of\_function(type1 arg1,type2 arg2, …. ,typen argn);

The most common approach is to declare the function at the beginning of the program.

**Function Call:**

The function can be accessed (i.e. called) by specifying its name, followed by a list of arguments enclosed in parenthesis and separated by commas. If function cal does not require any arguments, an empty pair of pair must follow the name of the function. The function call may be part of a simple expression ( such as assignment statement), or it may be one of the operands within a more complex expression. The arguments appearing in the function call are referred to as actual arguments, in contrast to the formal arguments that appear in the first line of the function definition.

**Function definition:**

The definition consist of a line called the declarator, followed by the function body. The function body is composed of the statements that make up the function, delimited by braces.

**Example**

#include<iostream.h>  
#include<conio.h>

int add(int a,int b); //function declaration

main()

{  
int sum;

sum=add(2,2); // function call

cout<<sum;

cout<<add(2,3); // function call

getch();

}  
int add(int a,int b) // function definition

{  
return a+b;

}

**OUTPUT:**



If you place the function definition before the main function, then there is no need of declaring the function. Function sum can be rewritten as

#include<iostream.h>  
#include<conio.h>  
int add(int a,int b) // function definition

{  
return a+b;

}  
main()  
{  
int sum,a=3,b=4;

sum=add(2,2); // function call

cout<<sum;

cout<<add(a,b); // function call

getch();

}



**Creating user define Header files:**

User can create the their own header files, by putting the user defined function in those header files. The header files that comes with the C compiler are placed in the include folder of the C compiler. So the header files created by the user should be placed in the include folder.

**Example**

Int add(int a,int b)

{  
return a+b;

}  
int sub(int a,int b)

{  
return a-b;

}  
 Save this program in the include folder of the tc compiler, i.e. c:\tc\include\myh.h, if TC compiler is installed on the C drive.

If you want to use this header file in your program, this can be done by simply placing the name of the header file in the beginning of the program.

**Example**

#include<iostream.h>  
#include<myh.h>  
main()  
{  
cout<<add(2,2);

cout<<sum(3,2);

}

**Lab Tasks:**

**Q#01** Develop a code which generates a user defined function that converts a given hexadecimal No. to a decimal integer Number.

**Q#02** Develop a code that finds the larger of two numbers entered from the keyboard. Use a function to do the actual comparison of the two numbers. Pass the two numbers to the function as arguments and function return the answer in main program.

**Q#03** State advantage of creating your own header file. Explain how you may call your header file from a c++ prog.

**Q#04** Develop a program in C++. to make your own a header file name mystuff.h in which the function of addition, subtraction, multiplication, division is included and then write a code that uses the same header file and performs the function accordingly.

**QUESTION NUMBER : 01:**

Develop a code which generates a user defined function that converts a given hexadecimal No. to a decimal integer Number.

**PROGRAM:**

#include <iostream>

using namespace std;

void decToHexa(int n)

{

char hexaDeciNum[100];

int i = 0;

while (n != 0) {

int temp = 0;

temp = n % 16;

if (temp < 10) {

hexaDeciNum[i] = temp + 48;

i++;

}

else {

hexaDeciNum[i] = temp + 55;

i++;

}

n = n / 16;

}

for (int j = i - 1; j >= 0; j--)

cout << hexaDeciNum[j];

}

int main()

{

int n = 3456;

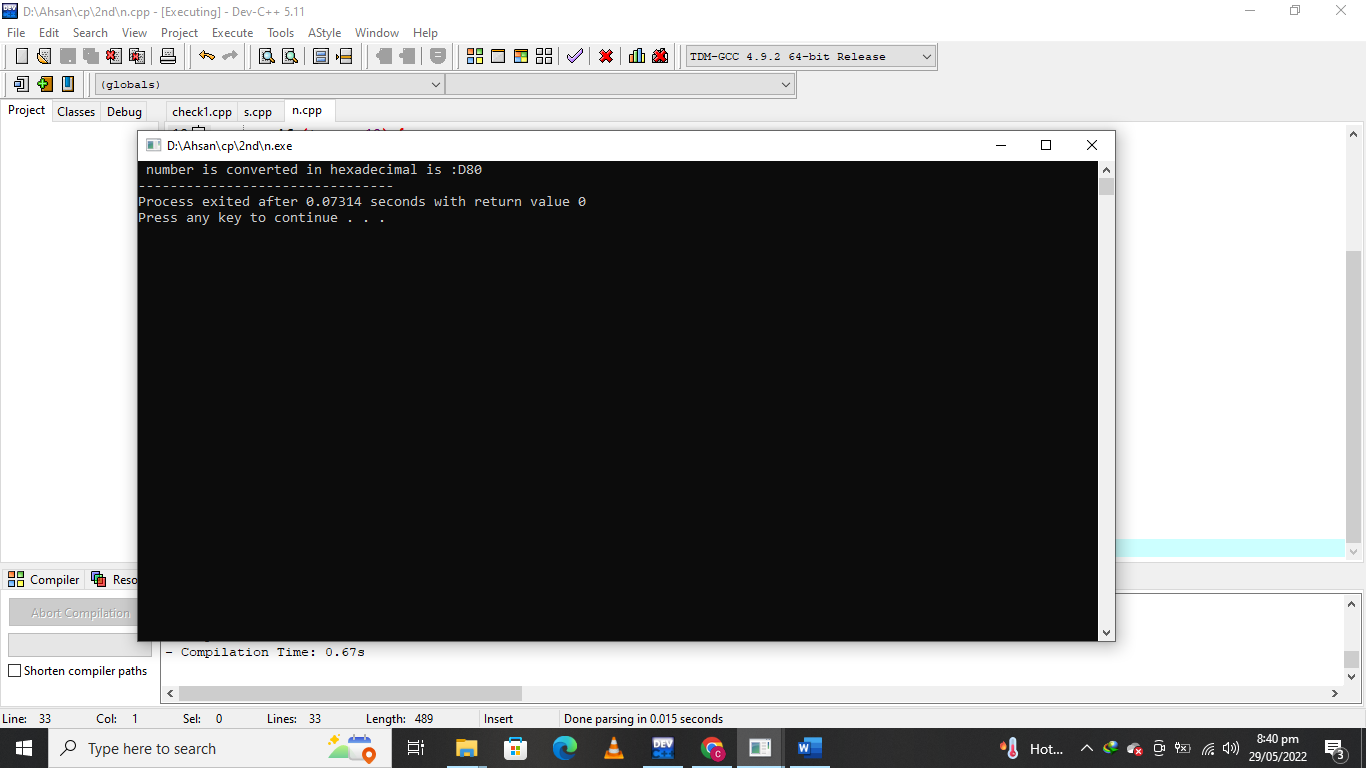
cout<<" number is converted in hexadecimal is :";

decToHexa(n);

return 0;

}

**OUTPUT:**



**QUESTION NUMBER : 02:**

Develop a code that finds the larger of two numbers entered from the keyboard. Use a function to do the actual comparison of the two numbers. Pass the two numbers to the function as arguments and function return the answer in main program.

**PROGRAM:**

#include <iostream>

#include <conio.h>

using namespace std;

int large(int num1,int num2);

main ()

{

int num1,num2;

cout <<"The program which we find which number is larger "<<endl;

cout<<large(num1,num2);

getch ();

}

int large(int num1,int num2)

{cout<<" Enter the first number :";

cin>>num1;

cout<<" Enter the second number :";

cin>>num2;

if(num1>num2)

cout<<"\nThe first number ("<<num1<<") is larger then the second ("<<num2<<").\n";

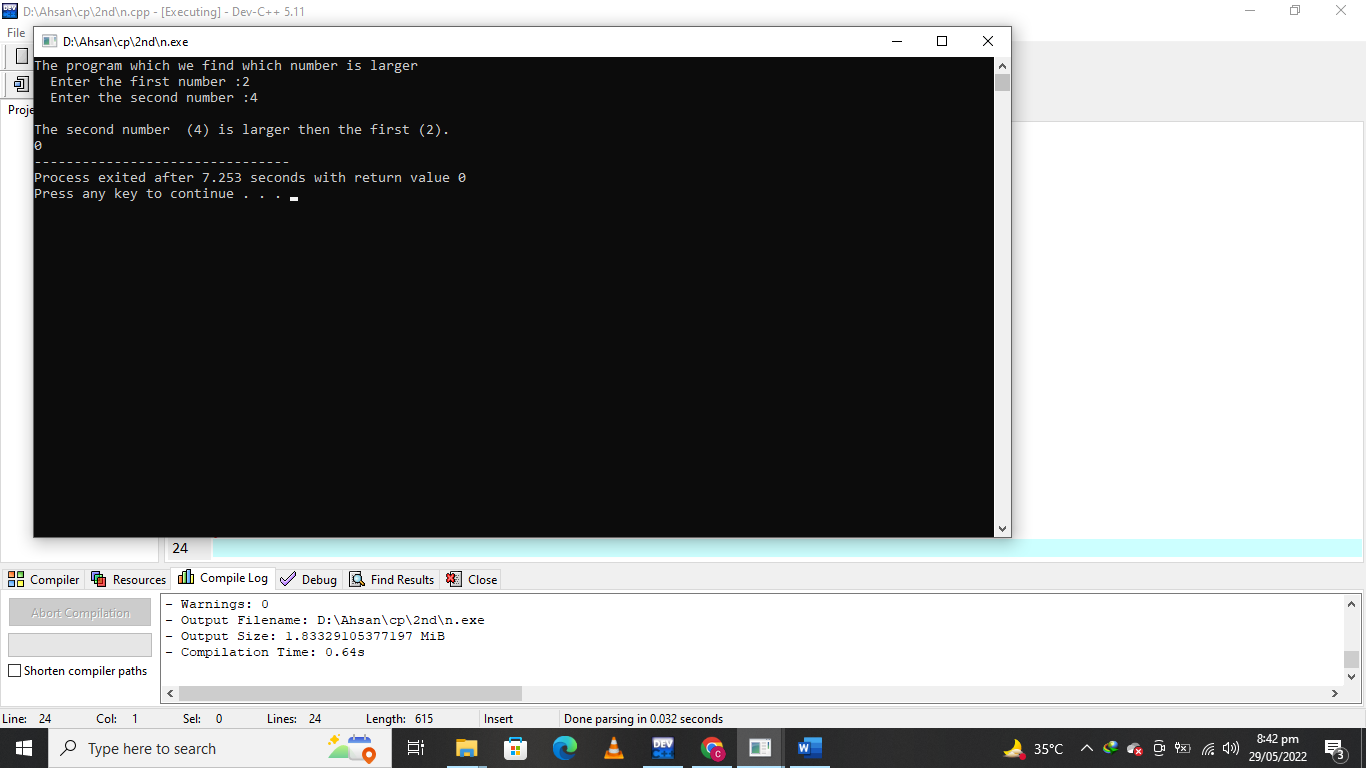
else

cout<<"\nThe second number ("<<num2<<") is larger then the first ("<<num1<<").\n";

return 0;

}

**OUTPUT:**



**QUESTION NUMBER : 03:**

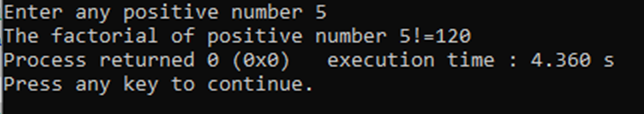
State advantage of creating your own header file. Explain how you may call your header file from a c++ prog.

**PROGRAM:**

|  |
| --- |
| **Header:**  int factorial(int num)  {  int i,fact=1;  for(i=1; i<=num; i++)  {  fact=fact\*i;  }  return fact;  } |

|  |
| --- |
| #include <iostream>  #include <conio.h>  #include "header.h"  using namespace std;  main()  {  int n;  cout<<"Enter any positive number ";  cin>>n;  cout<<"The factorial of positive number "<<n<<"!=" <<factorial(n);  return 0;  } |

**OUTPUT:**



**QUESTION NUMBER : 04:**

Develop a program in C++. to make your own a header file name mystuff.h in which the function of addition, subtraction, multiplication, division is included and then write a code that uses the same header file and performs the function accordingly.

**PROGRAM:**

|  |
| --- |
| **Header:**  using namespace std;  void cal(int a,int b ,char c)  {  if(c=='+')  cout <<a<<"+"<<b<<"="<<a+b<<endl;  else if(c=='-')  cout <<a<<"-"<<b<<"="<<a-b<<endl;  else if(c=='\*')  cout <<a<<"x"<<b<<"="<<a\*b<<endl;  else if(c=='/')  cout <<a<<"/"<<b<<"="<<a/b<<endl;  else  cout<<"Invalid operator"<<endl;  } |
| #include <iostream>  #include <conio.h>  #include "myshuff.h"  using namespace std;  main()  {  int x,y;  char c;  couot<<"Enter first number , operator and second number :";  cin>>x>>c>>y;  cal(x,y,c);  getch;  } |

**OUTPUT:**

