

Lab Session 1

Task 1:

Write a program to print text in following pattern,

```
Hello World
  Hello World
Hello World
```

```
#include <iostream>

using namespace std;

int main()
{
    cout << "Hello world \n\t Hello World \n Hello World" << endl;
    return 0;
}
```

Lab Session 2

Task 1:

How to insert single line and multiline comments in a C++ program.

- To print the single line comment we have to insert `“//”` before the line.
- For Multi-line comment we have to use `“/*”` and `“*/”` before and after the comments.

Task 2:

Variable Declarations can appear almost anywhere in the body of C++ function (T/F). If true, then discuss the situation in which variable declaration must be done prior to some specific task. Support your answer by giving example.

True. However, the right kind of variable should be declared earlier than both storing information in it or using it. Like, "int" for easy addition or subtraction, "bool" for proper or false, "float" or "double" for fractions.

For e.g.

```
#include <iostream>

using namespace std;

int main ()
{
    int a;
    double b,c;
    cin>>a>>b;
    c=a*b;
    cout<<c;
}
```

Task 3:

Calculate the maximum and minimum number that can be accommodated by int data type (calculate range).

It is calculated with the formula,

$$-(2^n-1) \text{ ----- } (2^n-1)-1$$

Where 'n' is the range of bits in record set, for an 'int' function. It is 32(4*8) bits.

So, the Range becomes:

$$-3,177,493,648 \text{ ----- } 3,177,493,647$$

Task 4:

What do you mean by Variable Declaration and Variable Definition in C/C++?

Declaration of a variable is for informing to the compiler the subsequent information: call of the variable, kind of value it holds and the preliminary value if any it takes. i.e., declaration offers information about the properties of a variable. Whereas, definition of a variable says wherein the variable receives stored.

Task 5:

Check the output of the following cout functions and write your comments.

1. `cout << "I am a computer geek, \rits a \blie."`
2. `cout << "a" << "\t" << "b" << "\t" << "c" <`

Several get away sequences are used on this code block.

First, the compiler writes "I am a computer geek", however then the "\r" repeats the line, so the compiler overwrites the preceding textual content with "Its a" then "\b" eliminates the ultimate space and writes "lie" instead.

Then, it writes "a", "b", "c" with "\t" including the specific range of areas in among three of them (eight areas).

Lab Session 3

Code 03

```
#include <iostream>

using namespace std;

int main()
{
    int number1 = 74, number2 = 82,
    number3 = 88;

    double average;

    average = number1 + number2 +
    number3 / 3;

    cout<<average;

    return 0;
}
```

Code 04

```
#include <iostream>

using namespace std;

int main()
{
    int number1 = 74, number2 = 82,
    number3 = 88;

    double average;

    average = (number1 + number2 +
    number3) / 3;

    cout<<average;

    return 0;
}
```

What did you observe from the output of the above two programs? Try to explain briefly.

Yes, in the first program, since the "/" function takes sequence so it is performed first. And in the second program "()" takes sequence instead, hence it is applied first. Thus, the difference in result.

Task 1:

Using compound assignment operators, write a program that generates the following output:

x = 2.5 y = 10

x = 25.0 y = 15

x = 250.0 y = 20

x = 2500.0 y = 25

Initialize x as float with value of 2.5 and y as int with value 10. In each successive stage, use *= operator for x and += operator for y to achieve the desired values.

```
#include <iostream>
#include <cmath>
#include <complex>

using namespace std;

int main()
{
    float x = 2.5 ;
    int y = 10 ;

    cout << "x = " << x ;
    cout << "\t\tty = " << y ;

    int count = 1 ;

    while (count <= 3){

        cout << "\nx = " << (x *= 10) << "                      y = " << (y += 5) ;

        count ++ ;

    }
    return 0;
}
```


Task 2:

Write a program that asks the user to enter the length of base and perpendicular of a right angle triangle. Then it determines the length of hypotenuse, angle between base and hypotenuse and angle between hypotenuse and perpendicular. Also find the sine and cosine values of these angles.

```
#include <iostream>
#include <cmath>
#include <complex>

using namespace std;
int main (void)
{
    double base , perp , hyp , sinTheta , cosTheta , Theta1 , Theta2 ;

    cout << "ENTER THE VALUE OF BASE: " << endl;
    cin >> base ;

    cout << "ENTER THE VALUE OF PERPENDICULAR: " << endl;
    cin >> perp ;

    hyp = sqrt((base*base)+(perp*perp));
    sinTheta = (perp/hyp);
    cosTheta = (base/hyp);
    Theta1 = asin (perp/hyp);
    Theta2 = acos (base / hyp);

    cout <<"THE HYPOTENUS IS: " << hyp << endl;
    cout <<"THE SINE OF ANGLE IS: " << sinTheta << endl;
    cout <<"THE COSINE OF ANGLE IS: " << cosTheta << endl;
    cout <<"THE ANGLE BETWEEN PERPENDICULAR AND HYPOTENUS IS: " << Theta1 << endl;
    cout <<"THE ANGLE BETWEEN BASE AND HYPOTENUS IS: " << Theta2 << endl;
    return 0;
}
```

Task 3:

Write a program that asks the user to enter coefficients a, b and c of the standard quadratic equation.

$$ax^2+bx+c=0$$

The program then should compute and display discriminant

$$|b^2 - 4ac|$$

And the roots of equation

$$x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a}$$

$$x_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$$

Finally, give opinion on how the program could be made more general to different input conditions

```

#include <iostream>
#include <cmath>
#include <complex>

using namespace std;
int main()
{
    int a , b , c;

    cout << "\tax2+bx+c=0"<<endl;

    cout <<"ENTER THE CO-EFFICIENT OF a: ";
    cin >> a ;

    cout <<"ENTER THE CO-EFFICIENT OF b: ";
    cin >> b ;

    cout <<"ENTER THE CO-EFFICIENT OF c: ";
    cin >> c ;

    cout << "\n\tFOR DISCRIMINANT: "<< endl;
    float dis = (b*b) - (4*a*c);
    cout << dis <<endl ;
    cout << "\n\tFOR ROOTS OF QUADRATIC EQUATION: "<< endl;
    if (dis > 0)
    {
        float x1 = ((-b) + sqrt((b*b)-(4*a*c)))/(2*a);
        float x2 = ((-b) - sqrt((b*b)-(4*a*c)))/(2*a);
        cout << "x1 = " << x1 << endl ;
        cout << "x2 = " << x2 << endl ;
    }
    else {
        cout << "DISCRIMINANT IS COMPLEX";
    }
    return 0;
}

```

Lab Session 4

Task 1 (Method 1):

Write a program that asks user to enter 3 numbers and then finds the largest and smallest among them and displays both largest and smallest number. This program can be written in many ways. Provide at-least two methods.

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

int main()
{ int x , y , z , largest , smallest ;
  cout<< "enter value of x:";
  cin >> x;
  cout<< "enter value of y:";
  cin >> y;
  cout<< "enter value of z:";
  cin >> z;
  largest = x;
  if (largest < y)
  {
    largest = y;
  }
  if (largest < z)
  {
    largest = z;
  }
  smallest = x;
  if (smallest > y)
  {
    smallest = y;
  }
  if (smallest > z)
  {
    smallest = z;
  }
  cout<<"largest:"<<largest<<"\n";
  cout<<"smallest:"<<smallest<<"\n";

  return 0;
}
```

Task 1 (Method 2):

Write a program that asks user to enter 3 numbers and then finds the largest and smallest among them and displays both largest and smallest number. This program can be written in many ways. Provide at-least two methods.

```
#include<iostream>
using namespace std;
int main ()
{
    int arr[10], i, max, min;
    int n = 3 ;
    cout << "Enter the elements of the array : ";
    for (i = 0; i < n; i++)
        cin >> arr[i];
    max = arr[0];
    for (i = 0; i < n; i++)
    {
        if (max < arr[i])
            max = arr[i];
    }
    min = arr[0];
    for (i = 0; i < n; i++)
    {
        if (min > arr[i])
            min = arr[i];
    }
    cout << "Largest element : " << max;
    cout << "\nSmallest element : " << min;
    return 0;
}
```

Task 2:

C++ provides an alternate approach for if () – else if () statements, that is switch () – case statement. Use literature and internet resources to understand using it. Then write a calculator program written in lab session using switch ()-case statements. If the user entered the operator other than +,-,*,/ then program should print “Invalid Operator” on screen.

```
#include <iostream>
#include <cmath>

using namespace std;

int main (void)
{
    double num1 , num2 , result;
    char op ;

    cout << "ENTER THE FIRST NUMBER: ";
    cin >> num1 ;

    cout << "ENTER THE OPERATION YOU WANT TO PERFORM: ";
    cin >> op ;

    cout << "ENTER THE SECOND NUMBER: ";
    cin >> num2 ;

    switch (op) {
    case '+': cout << "RESULT = " << num1 + num2 << endl;
    break ;

    case '-': cout << "RESULT = " << num1 - num2 << endl;
    break ;

    case '*': cout << "RESULT = " << num1 * num2 << endl;
    break ;

    case '/': cout << "RESULT = " << num1 / num2 << endl;
    break ;

    default : cout << "\nINVALID OPERATOR\n-----\n";
    break ;
    }
    return 0;
}
```

Lab Session 5

Task 1(a):

Write a program to print following pattern using for loops (do not use if, if-else or any other decision making statement)

```
*
***
*****
*****
```

```
#include <iostream>

using namespace std;

int main()
{
    int num_of_spaces = 3;
    int num_of_stars = 1;

    for (int i = 1; i<=4; i++){

        for (int j = 1; j<= num_of_spaces; j++){
            cout <<" ";
        }

        for (int k = 1; k<= num_of_stars; k++){
            cout <<"*";
        }

        cout << endl;
        num_of_spaces = num_of_spaces - 1;
        num_of_stars = num_of_stars + 2;
    }
    return 0;
}
```

Task 1(b):

Write a program to print following pattern using for loops (do not use if, if-else or any other decision making statement).

```

1
121
12321
1234321

```

```

#include <iostream>

using namespace std;

int main()
{
    int i,j,row;

    for(i=1;i<=4;i++)
    {
        for(j=1;j<=4-i;j++)
        {
            cout << " ";
        }

        for(j=1;j<=i;j++)
        {
            cout << j ;
        }

        for(j=i-1;j>=1;j--)
        {
            cout << j ;
        }
        cout << "\n";
    }

    return 0;
}

```

Task 2:

Using for () loops, write a program that displays all possible combination of 6 bit binary number. (Hint: You shall need 6 int variables for the six digits).

```
#include <iostream>

using namespace std;

int main ()
{
    int num1 , num2 , num3 , num4 , num5 , num6 ;

    for (num1 = 0 ; num1 <= 1 ; num1++){
        for (num2 = 0 ; num2 <= 1 ; num2++){
            for (num3 = 0 ; num3 <= 1 ; num3++){
                for (num4 = 0 ; num4 <= 1 ; num4++){
                    for (num5 = 0 ; num5 <= 1 ; num5++){
                        for (num6 = 0 ; num6 <= 1 ; num6++){
                            cout <<"\n"<<num1<< num2<< num3 << num4 << num5 <<
num6 ;
                        }
                    }
                }
            }
        }
    }

    return 0;
}
```


Lab Session 6

Task 1:

Write a program that continuously asks user to enter an integer and displays the SUM of the current input with all previous input. The program continuous to run until the SUM value is less than equal to 100. Use while () loop.

```
#include <iostream>

using namespace std;

int main ()
{
    double userInput , sum ;
    sum = 0 ;

    while (sum!=100)
    {

        cout << "ENTER A NUMBER OF YOUR DESIRE: " ;
        cin >> userInput ;

        sum = sum + userInput ;

        cout <<"RUNNING SUM: "<< sum << "\n" ;

        if(sum >= 100)
        {
            cout << "SUM EXCEEDS 100 \nPROGRAM TERMINATED\n-----";
            break ;
        }
    }
}
```

Task 2:

Write a program that counts number of digits in an integer entered by the user. Use while () loop.

```
#include<iostream>

using namespace std;

int main()
{
    int userInput ;
    int count = 0;

    cout << "ENTER ANY NUMBER YOU WISH: ";
    cin >> userInput;

    while (userInput > 0)
    {
        userInput = userInput / 10;

        count++;
    }
    cout << "\nNUMBER OF DIGITS IN YOUR INPUTED NUMBER IS: " << count << "\n" ;
    return 0;
}
```

Lab session 7

Task 1:

Q1: Halley's Method for Determination of roots of polynomial is

$$x_{n+1} = x_n - \frac{2f(x_n)f'(x_n)}{2f'(x_n)^2 - f(x_n)f''(x_n)}$$

Write a program to find roots of the following polynomial using Halley's method, precision is 0.0001

$$f(x) = ax^3 + bx^2 + cx + d$$

```
#include<iostream>
#include<cmath>
Using namespace std;
Int main()
{
    Int a,b,c,d;
    Float Xn;
    Double fxn,fpnx,fdxn,lnG;
    Cout<<"Enter the values of coefficients of the equation"<<endl;
    Cin>>a>>b>>c>>d;
    Cout<<"Your equation is"<<"="<<endl;
    Cout<<a<<"x^3+"<<b<<"x^2+"<<c<<"x+"<<d<<endl;
    Cout<<"Enter some initial guess"<<"="<<endl;
    Cin>>lnG;
    Xn=lnG/2;
    Fxn=(a*Xn*Xn*Xn)+(b*Xn*Xn)+(c*Xn)+d;
    Fpxn=(3*a*Xn*Xn)+(2*b*Xn)+c;
    Fdxn=(6*a*Xn)+(2*b);
    While(fabs((2*fxn*fpnx)/((2*pow(fpnx,2))-(fxn*fdxn)))>0.0001)
    {
        Fxn=(a*Xn*Xn*Xn)+(b*Xn*Xn)+(c*Xn)+d;
        Fpxn=(3*a*Xn*Xn)+(2*b*Xn)+c;
        Fdxn=(6*a*Xn)+(2*b);
        Xn=Xn-(2*fxn*fpnx)/((2*pow(fpnx,2))-(fxn*fdxn));
    }
    Cout<<"The root of the equation becomes"<<"="<<Xn<<endl;
    Return 0;
}
```

Task 2:

Q2: Write an iterative algorithm to implement the following expansion (precision upto 0.0001)

$$\sin x = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n+1)!} x^{2n+1} = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots \quad \text{for all } x$$

```
#include <iostream>
Using namespace std;
Double radians(double degrees) // converts degrees to radians
{
    Double radians;
    Double const pi = 3.14159265358979323846;
    Radians = (pi/180)*degrees;
    Return radians;
}
Double factorial(int x) //calculates the factorial
{
    Double fact = 1;
    For(; x >= 1 ; x--)
    {
        Fact = x * fact;
    }
    Return fact;
}
Double power(double x,double n) //calculates the power of x
{
    Double output = 1;
    While(n>0)
    {
        Output =( x*output);
        n--;
    }
    Return output;
}
Float sin(double radians) //value of sine by Taylors series
{
    Double a,b,c;
    Float result = 0;
    For(int y=0 ; y!=9 ; y++)
    {
        A= power(-1,y);
        B= power(radians, (2*y)+1);
        C= factorial((2*y)+1);
        Result = result+ (a*b)/c;
    }
    Return result;
}
Double n,output;
Int main()
```

```
{  
    Cout<<"enter the value\t";  
    Cin>>n;  
    N = radians(n);  
    Cout<< "\nthe value in radians is\t"<< n << "\n";  
    Output = sin(n);  
    Cout<< "\nsine of the given value is\t"<< output;  
    Return 0;  
}
```

Lab session 8

Task 1:

Read in 20 numbers in an array, each of which is in between 10 and 100 – if the number is not in this range, ask user to re-enter. As each number is read by the program, print it only if it is not a duplicate of a number already read

```
#include <iostream>
Using namespace std;
Void checkIfExists(int b[], int index)
{
    For (int i = 0; i < index; i++)
    {
        If (b[index] == b[i])
        {
            Return;
        }
    }
    Cout <<b[index]<< " ";
}
Int main()
{
    Int i;
    Int numArray[20] = { 0 };
    Cout << "Enter "<<20<<" numbers: ";

    For (i = 0; i<20; i++)
    {
        Cin >> numArray[i];
    }
    For(i=0;i<20;i++)
    {
        If(numArray[i]>=10 && numArray[i]<=100)
        {
            Cout << "You input: \n";

            For (i = 0; i<20; i++)
            {
                Cout <<numArray[i]<<" ";
            }
            Cout << "\n";

            Cout << "The nonduplicate values are: \n";
            Cout << numArray[0]<<" ";

            For (i = 1; i<20; i++)
            {
                checkIfExists(numArray, i);
            }
        }
    }
}
```

```
        }  
        Else  
        {  
            Cout << "\n\nPlease input in range between 10  
to 100 only ";  
            Break;  
        }  
    }  
    Return 0;  
}
```

Task 2:

Write a simple database program that stores name, roll no., and cgpa in FE , all in separate arrays, for 25 students. The program should be able to let the user enter records, display records and replace any one of the records (switch()-case can be used to give these options to user). The program must continue until ESC is pressed. [Note: A single 'record' means name, roll no., and cgpa of one student]..

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

Int main(){
    Int roll[25];
    String name[25];
    Float cgpa[25];
    Int n;
    Do{

        Cout<<endl<<"Menu"<<endl<<"1-Enter records"<<endl<<"2-Display
Records"<<endl<<"3-Replace Records"<<endl<<"0-Exit";
        Cin>>n;
        Switch(n){
            Case 1:
                For(int i=0;i<25;i++){
                    Cout<<endl<<"Enter the name of student
"<<i+1<<" : ";
                    Cin>>name[i];

                    Cout<<endl<<"Enter the GPA of student "<<i+1<<"
: ";
                    Cin>>cgpa[i];

                    Cout<<endl<<"Enter the roll number of student
"<<i+1<<" : ";
                    Cin>>roll[i];
                }
                Break;
            Int number;
            Case 2:
                Cout<<endl<<"Enter the roll number to display the
record of the student: ";
                Cin>>number;
                For(int i=0;i<25;i++){
                    If(number == roll[i]){
                        Cout<<endl<<"Name: ";
                        Cout<<name[i];

                        Cout<<endl<<"Roll Number: ";
                        Cout<<roll[i];

                        Cout<<endl<<"Name: ";
                        Cout<<cgpa[i];
                    }
                }
            }
        }
    }
}
```



```

        Else{
            If(i<25){
                Continue;
            }
            Else{
                Cout<<"Record not found!";
            }
        }

    }

    Break;

    Case 3:
    Cout<<"Enter the roll number to replace the student
details: ";
    Cin>>number;
    Cout<<endl<<endl<<"Replacing the Details.....";
    For(int i =0;i<25;i++){
        If(number == roll[i]){
            Cout<<endl<<endl<<"Replacing the
Details.....";

            Cout<<endl<<"Name: ";
            Cin>>name[i];

            Cout<<endl<<"GPA: ";
            Cin>>cgpa[i];

        }
        Else{
            If(i<25){
                Continue;
            }
            Else{
                Cout<<"Record not found!";
            }
        }
    }

    Break;

}
}while(n!=0);
Return 0;
}

```

Lab session 9

Task 1:

Write program with a function that accepts 3 int type numbers and returns the smallest among them. The function is called minimum().

```
#include <iostream>
Using namespace std;
Int minimum(int a , int b , int c);
Int a,b,c;
Int main (void)
{
    Int a,b,c;
    Cout<<"Enter the three integers ";
    Cin>>a>>b>>c;
    Cout<<"The smallest or minimum number is "<< minimum( a, b ,c);
    Return 0;}

Int minimum (int a, int b , int c)
{
    If(a<b && a<c)
    {return a;}
    Else if(b<a && b<c)
    {return b;}
    Else(c<a && c<b);
    {return c;}
}
```

Task 2:

Write a void function that generates a precise delay of 2 seconds whenever it is called. The function should contain clock() function or time() function from ctime, for precise timing.

```
#include <iostream>

#include <string.h>

#include <ctime>

#include <windows.h>

Using namespace std;

Void time_delay()

{

    Time_t my_time = time(NULL);

    Cout<<("%s", ctime(&my_time));

    Sleep(2000);

    Time_t y_time = time(NULL);

    Cout<<("%s", ctime(&y_time));

    Clock_t now;

    Now = clock();

    Cout << "Processor took " << now/1000;

};

Int main(){

    Time_delay();

    Return 0;

}
```

Lab session 10

Task 1:

Write a recursive function to implement Newton Raphson Method algorithm to determine square root of a number.

```
Include<iostream>
#include<cmath>
Using namespace std;
Float N,low,high,root;
Float square_root(float root);
Int main(void)
{
    Cout<<"Enter a +ve number to calculate its square root:";
    Cin>>N;
    Cout<<"\n square root of "<<N<<" = "<<square_root(root);
    Return 0;
}
Float square_root(float root)
{
    If (N<0)
    {
        Return 0;
    }
    Else
    {
        Root=N/2.0;
        While (fabs(((root*root)-N))>0.0001)
        {
            Root=root-((root*root-N)/(2.0*root));
        }
    }
    Return root;}
```

Task 2:

Write a recursive function to find Greatest Common Divisor of two numbers using Euclid Remainder Algorithm

```
#include<iostream>

#include<conio.h>

#include<stdlib.h>

Using namespace std;

Int gcd(int u, int v)

{

    Return (v != 0) ? gcd(v, u % v) : u;

}

Int main(void)

{

    Int number1, number2, result;

    Cout << "Enter two numbers to find GCD using Euclidean
algorithm: ";

    Cin >> number1 >> number2;

    Result = gcd(number1, number2);

    If (gcd)

        Cout << "\nThe GCD of " << number1 << " and " << number2
<< " is: " << result

            << endl;

    Else

        Cout << "\nInvalid input!!!\n";

    Return 0;

}
```

Task 3:

Test for a number if it is prime or composite using recursion

```
#include <iostream>
Using namespace std;
Int prime_or_composite(int num);
Int i;
Int main()
{
    Int num;
    Cout << "Enter a number: ";
    Cin >> num;
    Cout << prime_or_composite(num) ;
    Return 0;
}
Int prime_or_composite(int num)
{
    For (int i = 2; i < num; i++)
    {
        If (num%i == 0 )
        {
            Cout << "The number is composite" << endl;
            Return 0 ;}
        }
    }
```

Task 4:

Write a recursive function to implement the following expansion (precision upto 0.0001)

$$\sin x = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n+1)!} x^{2n+1} = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots \quad \text{for all } x$$

```
#include<iostream>
#include<cmath>
#include<iomanip>
#include<conio.h>
Using namespace std;
Long long factorial (int y);
Int main ()
{
Double x, sin = 0;
Int a;
Cout<<"Enter a value for sine evaluation in radians:"<<endl;
Cin>>x;
Cout<<"Enter number of terms in the summation : "<<endl;
Cin>>a;
Int n = 0;
While (n<a)
{sin=sin+(((pow(-1, n))/factorial(2*n+1))*pow(x,2*n+1));
N++;}
Cout<<"Result : "<<setprecision(10)<<sin<<endl;
Getch();
}
Long long factorial (int y)
{ long long fact=1;
While (y>0)
{fact=fact*y; y--;}
Return fact;}
```

Lab session 11

Task 1:

Selection sort algorithm can be used to sort an array in ascending order. The first iteration of the algorithm selects the smallest element in the array and swaps it with the first element. The second iteration selects the second-smallest element (which is the smallest element of the remaining elements) and swaps it with the second element. The algorithm continues until the last iteration selects the second-largest element and swaps it with the second-to-last index, leaving the largest element in the last index. As an example, consider the array 34 56 4 10 77 51 93 30 5 52. A program that implements the selection sort first determines the smallest value (4) in the array, which is contained in element 2. The program swaps the 4 with the value in element 0 (34), resulting in 4 56 34 10 77 51 93 30 5 52. The program then determines the smallest value of the remaining elements (all elements except 4), which is 5, contained in element 8. The program swaps the 5 with the 56 in element 1, resulting in 4 5 34 10 77 51 93 30 56 52. On the third iteration, the program determines the next smallest value, 10, and swaps it with the value in element 2 (34). 4 5 10 34 77 51 93 30 56 52. The process continues until the array is fully sorted. 4 5 10 30 34 51 52 56 77 93. Using pass by reference feature of pointers, implement the selection sort algorithm.

```
#include<iostream>

#include<iomanip>

Using namespace std;

Void selectionSort( int *const array, const int size );

Void swap( int *const element1Ptr, int *const element2Ptr );

Int main()
{
    Const int arraySize = 10;

    Int a[arraySize] = {34,56,4,10,77,51,93,30,5,52};

    Cout<<"The unsorted array is"<<endl;

    For ( int j=0; j<arraySize; j++ )
    {
        Cout<<"Element at location "<<j<<" : "<<setw(4)<<a[j];

        Cout<<endl;
    }

    Cout<<endl<<endl;

    selectionSort(a,arraySize);

    cout << "The sorted array is:"<<endl;

    for ( int j=0; j<arraySize; j++ )
```



```
{
Cout<<"Element at location "<<j<<" : "<<setw(4)<<a[j];
Cout<<endl;
}
Return 0;
}

Void selectionSort( int *const array, const int size )
{
Int smallest;
For ( int i = 0; i < size - 1; i++ )
{
Smallest=i;
For ( int index=i+1; index<size; index++ )
If (array[index]<array[smallest])
Smallest=index;
Swap(&array[i],&array[smallest]);
}
}

Void swap(int *const element1Ptr, int *const element2Ptr )
{
Int hold = *element1Ptr;
*element1Ptr = *element2Ptr;
*element2Ptr = hold;
}
```

Task 2:

Explore all methods of viewing addresses of variables. Also explore address storing mechanisms (pointer variables). Finally use this knowledge to access and manipulate arrays and call multiple variables from functions using pointers.

--

Lab session 12

Task 1:

Develop a simple text editor application with File, Edit and Fonts Menus. It should be able to create new files, display previously stored text files, edit files and save any changes.