**Experiment#14**

**OBJECTIVE:**

**To become familiar with pointers and arrays & pointers and Functions**

**Pointers:**

The idea behind pointers is not complicated as we know that every byte in the computer memory has an address. These addresses are known as pointer constants. Normally a variable directly contains a specific value where as a pointer on the other hand contains the address of a variable that contains a specific value.

We can find the address occupied by a variable by using the address of operator as illustrated in the following example.

**Example:**

#include <constream>

void main()

{int age;

cout<<”Enter your age: “;

cin>>name;

cout<<”\nThe address your age in the memory is “<<&age;

getche();}

A variable that holds an address value is called a pointer variable. We can initialize a pointer variable as we initialize a variable but difference is that we put \* sign after the data type.

Int\* ptr;

The \* means pointer to. Thus the statement defines the variable ptr as a pointer to integer.

Pointers like any other variables must be declared before they can be used. They can be initialized to 0, null or an address.

**Pointers & Arrays**

There is a close association between pointers & arrays. We can access array elements by using the array notation or by using the pointer notation. For example:

#include <constream>

void main()

int array[5]=(1,5,2,0,4);

for (int j=0;j<5;j++)

{cout<<array[j];

cout<<\*(array+j)<<endl;}

getche();}

The expression \*(array+j) has the same effect as array[j] & the output of both the statements is same because the name of the array is its address.

**Pointers & Functions**

There are two ways to pass arguments to a function i.e. by value, by reference & by pointer. If the function is intended to modify variables in the calling program then these variables cannot be passed value by through either a reference argument or a pointer can be used in this situation.

Example:

***// Program 1***

#include <iostream>

int days(int &);

void main()

{int age;

cout<<"\n Enter your age: ";

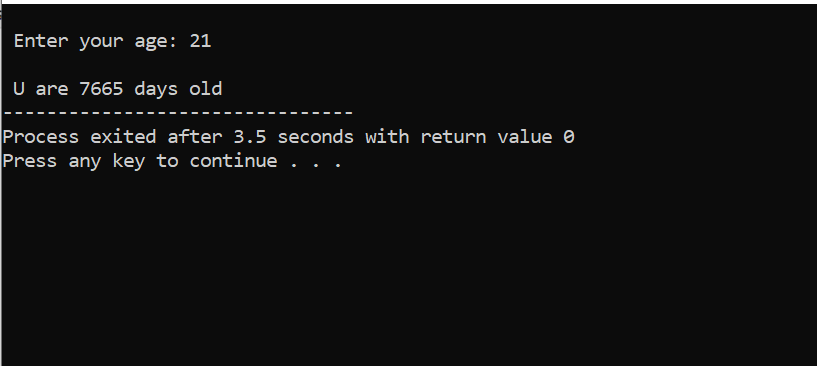
cin>>age;

cout<<"\n U are "<<days(age)<<" days old";

getche();}

int days(int & y)

{return(365\*y);}



***// Program 2***

#include <iostream>

const int n=3;

void main()

{void addarray(double\*,double\*,double\*,int) ;

double A[n]={10.0,2.3,20.2};

double B[n]={50.0,23.0,23.02};

double result[n];

addarray(A,B,result,n) ;

cout <<"the elements of array is";

cout <<"{";

for (int i=0;i<n;i++)

{cout <<"\t"<< \*(result+i)<<",";}

cout<<"}";

getche();}



void addarray(double\* a,double\* b,double\* c,int x)

{for (int i=0;i<x;i++)

\*(c+i)=\*(a+i)+\*(b+i);}

In program 2, we have added 2 initialized arrays by using the pointer notation as we can see from the statement void addarray(double\*,double\*,double\*,int)we have given the data types with \* signs. The reason is that we are by passing arrays as the arguments & array name has the address of the array.

**Lab Tasks:**

QNo:01 Display the outputs of above codes.

QNo:02 Develop a C++ code that uses pointer type argument and passes these argument to user defined function and then function returns the value in main program.

**QUESTION NUMBER : 01:**

Display the outputs of above codes.

**ANSWER:**

You can find your answer in above programs.

**QUESTION NUMBER : 02:**

Develop a C++ code that uses pointer type argument and passes these argument to user defined function and then function returns the value in main program.

**PROGRAM:**

#include <iostream>

using namespace std;

void swap(int &n1, int &n2) {

int temp;

temp = n1;

n1 = n2;

n2 = temp;

}

int main()

{

int a = 1, b = 2;

cout << "Before swapping" << endl;

cout << "a = " << a << endl;

cout << "b = " << b << endl;

swap(a, b);

cout << "\nAfter swapping" << endl;

cout << "a = " << a << endl;

cout << "b = " << b << endl;

return 0;

}

**OUTPUT:**

