**Experiment: 9**

PART A

(PART A: TO BE REFFERED BY STUDENTS)

**Aim:** **To study pointers in C++.**

**Learning Outcomes: Learner would be able to**

Demonstrate the use of pointers

**Tasks:**

1. WAP to demonstrate different arithmetic operations on pointers.
2. WAP to swap values of two variables using pointers.
3. WAP to largest of the elements of an array using a function and passing argument using pointer.
4. WAP to demonstrate use of a pointer to pointer.
5. WAP to demonstrate string handling using pointer.

**Theory:**

A pointer is a variable whose value is the address of another variable. Like any variable or constant, you must declare a pointer before you can work with it. The general form of a pointer variable declaration is −

type \*var-name;

Here, type is the pointer's base type; it must be a valid C++ type and var-name is the name of the pointer variable. The asterisk you used to declare a pointer is the same asterisk that you use for multiplication. However, in this statement the asterisk is being used to designate a variable as a pointer. Following are the valid pointer declaration −

int \*ip; // pointer to an integer

double \*dp; // pointer to a double

float \*fp; // pointer to a float

char \*ch // pointer to character

The actual data type of the value of all pointers, whether integer, float, character, or otherwise, is the same, a long hexadecimal number that represents a memory address. The only difference between pointers of different data types is the data type of the variable or constant that the pointer points to.

Using Pointers in C++

There are few important operations, which we will do with the pointers very frequently. (a) We define a pointer variable. (b) Assign the address of a variable to a pointer. (c) Finally access the value at the address available in the pointer variable. This is done by using unary operator \* that returns the value of the variable located at the address specified by its operand. Following example makes use of these operations −

Live Demo

#include <iostream>

using namespace std;

int main () {

int var = 20; // actual variable declaration.

int \*ip; // pointer variable

ip = &var; // store address of var in pointer variable

cout << "Value of var variable: ";

cout << var << endl;

// print the address stored in ip pointer variable

cout << "Address stored in ip variable: ";

cout << ip << endl;

// access the value at the address available in pointer

cout << "Value of \*ip variable: ";

cout << \*ip << endl;

return 0;

}

When the above code is compiled and executed, it produces result something as follows −

Value of var variable: 20

Address stored in ip variable: 0xbfc601ac

Value of \*ip variable: 20

PART B

(PART B: TO BE COMPLETED BY STUDENTS)

Students must submit the soft copy as per following segments within two hours of the practical. The soft copy must be uploaded on the portal at the end of the practical. The filename should be **PPS\_batch\_rollno\_experimentno Example: PPS\_B2\_B001\_Exp1**

|  |  |
| --- | --- |
| **Roll No.:** | **Name:** |
| **Prog/Yr/Sem:** | **Batch:** |
| **Date of Experiment:** | **Date of Submission:** |

**Task 1:**

**Task 2:**

**Task 3:**

**Task 4:**

**Task 5:**

**Conclusion (Learning Outcomes):** Reflect on the questions answered by you jot down your learnings about the Topic: Structure and Union

**Home Work Questions:**

**What will be the output of the following program?**

|  |
| --- |
| #include <iostream>  **using** **namespace** std;    **int** main()  {  **int** a = 32, \*ptr = &a;  **char** ch = 'A', &cho = ch;        cho += a;      \*ptr += ch;      cout << a << ", " << ch << endl;  **return** 0;  } |