|  |  |
| --- | --- |
| download | COMSATS University Islamabad, Vehari Campus Department of Computer Science |

**Name: Zubair Farooq**

**Class: BCS-SP22-4A Submission Deadline: 10 Sep 2023**

**Subject: Data Structures and Algorithms-Lab Instructor: Yasmeen Jana Max Marks: 10 Reg. No: SP22-BCS-004(A)**

**Email:** [**yasmeenjana@cuivehari.edu.pk**](mailto:yasmeenjana@cuivehari.edu.pk)

**You can ask queries related to Lab Activities on the above email.**

**Activity 1:**

Create a GitHub Account. Make a repository with the name “**DSA\_Lab”. Mention the link here after the account creation.**

**Solution:**

**https://github.com/ZubairFarooq004/-DSA\_Lab-.git**

**Activity 2:**

Write any 15 programs that will explain the concepts of pointers.

In this file, you should place the code and its output screenshot.

After completing the activities, Upload the final pdf and code to the “**DSA\_Lab”** repository.

**Name: Zubair Farooq**

**Roll no: SP22-BCS-004(A)**

**Assignment: 01**

**Subject: Data Structures**

**Teacher: Mam Yasmeen**

**CUI Vehari**

**Write 15 codes that explain the concepts of pointers.**

(1)

Pointer declaration and initialization:

#include<iostream>

using namespace std;

int main() {

int num = 42;

int\* ptr = &num;

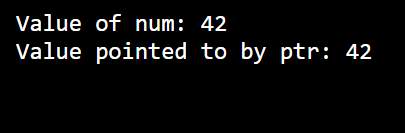
cout << “Value of num: ” << num << endl;

cout << “Value pointed to by ptr: ” << \*ptr << endl;

return 0;

}

Output:



(2)

Pointer Arithmetic:

#include<iostream>

using namespace std;

int main (){

int arr[] = {1,2,3,4,5};

int\* ptr = arr;

cout << “ First element: ” << \*ptr << endl;

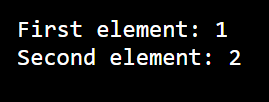
ptr++;

cout << “ Second element: ” << \*ptr << endl;

return 0;

}

Output:



(3)

Passing pointers to by function:

#include<iostream>

using namespace std;

void modifyValue(int\* ptr) {

\*ptr = 99;

}

int main (){

int num = 42;

cout << “Before: ” << num <<endl;

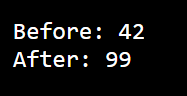
modifyValue (&num);

cout << “After: ” << num <<endl;

return 0;

}

Output:



(4)

Pointer to pointer (Double pointer):

#include<iostream>

using namespace std;

int main() {

int num = 42;

int\* ptr1 = &num;

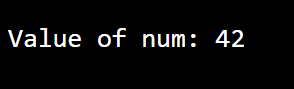
int\*\* ptr2 = &ptr1;

cout << “Value of num: ”<< \*\*ptr2 << endl;

return 0;

}

Output:



(5)

Dynamic Memory Allocation (malloc):

#include <iostream>

using namespace std;

int main() {

int\* ptr = ( int\*)malloc(sizeof(int));

\*ptr = 77;

cout << “Value : ” <<\*ptr << endl;

free (ptr);

return 0;

}

Output:



(6)

Dynamic memory allocation(new and delete):

#include <iostream>

using namespace std;

int main() {

int\* ptr = new int;

\* ptr = 88;

cout << “Value : ” <<\*ptr << endl;

delete (ptr);

return 0;

}

Output:



(7)

Array of Pointers:

#include <iostream>

using namespace std;

int main (){

int num1 = 10, num2 = 20, num3 = 30;

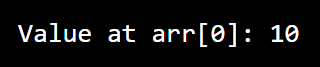
int\* arr[] = {&num1,&num2,&num3};

cout << “ Value at arr[0]: ” << \*arr[0] << endl;

return 0;

}

Output:



(8)

Pointer to Function:

#include <iostream>

using namespace std;

int add(int a,int b){

return a + b;

}

int main (){

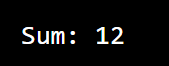
int (\* ptr)(int , int) = add;

cout << “ Sum: ” << ptr(5,7) << endl;

return 0;

}

Output:



(9)

Pointer to constant Data:

#include <iostream>

using namespace std;

int main() {

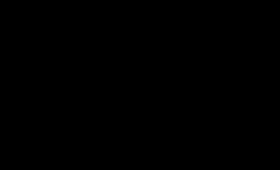
const int num = 42;

return 0;

}

Output:

Program didn,t output anything



(10)

Constant Pointer:

#include <iostream>

using namespace std;

int main() {

int num = 42;

int\* const ptr = &num;

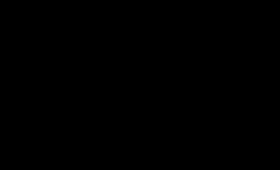
\*ptr = 50;

return 0;

}

Output:

Program didn,t output anything



(11)

Pointer to const pointer:

using namespace std;

int main() {

int num = 42;

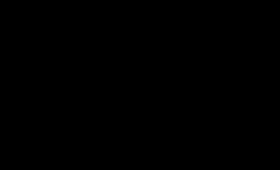
const int\* const ptr = &num;

return 0;

}

Output:

Program didn,t output anything



(12)

Pointer to Function Pointer:

#include <iostream>

using namespace std;

int add(int a,int b){

return a + b;

}

int main (){

int (\* ptr)(int , int) = add;

int (\*\* pptr)(int , int) = &ptr;

cout << “ Sum: ” << (\*pptr)(5,7) << endl;

return 0;

}

Output:



(13)

Pointers and Arrays:

#include <iostream>

using namespace std;

int main (){

int arr[] ={ 1,2,3,4,5};

int\* ptr = arr;

for (int i = 0; i < 5; i ++){

cout << “ Element: ” << i << “:” << \*ptr << endl;

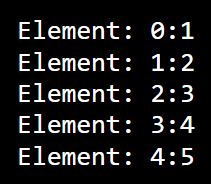
ptr ++;

}

return 0;

}

Output:



(14)

Pointer to member function:

#include <iostream>

using namespace std;

class MyClass {

public:

void display() {

cout << "Hello from MyClass!" << endl;

}

};

int main() {

void (MyClass::\*ptr)() = &MyClass::display;

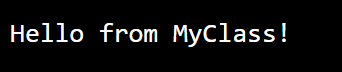
MyClass obj;

(obj.\*ptr)();

    return 0;

}

Output:



(15)

Pointers and Strings:

#include <iostream>

using namespace std;

int main() {

char str[] = "Hello, Pointers!";

char\* ptr = str;

while (\*ptr != '\0') {

cout << \*ptr;

ptr++;

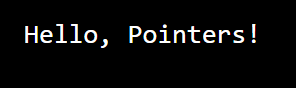
}

cout << endl;

    return 0;

}

Output:



Thank you!