**Practical:-13 Write a program to demonstrate use of virtual function.**

**Name:-Pranav Rajesh Tayade**

**Roll No:-153**

**Class:-FY BCA(B5)**

**DIV:-B**

#include<iostream>

using namespace std;

class sample

{

public:

virtual void display()

{

cout<<"\nsample class display";

}

};

class Test:public sample

{

public:

void display()

{

cout<<"\n Test class display";

}

};

int main()

{

sample \*p;

Test obj;

p=&obj;

p->display();

return 0;

}

**Output:**

Test class display

**Practical:-14(a) Write a program to demonstrate the use of a pointer to pointer.**

**Name:-Pranav Rajesh Tayade**

**Roll No:-153**

**Class:-FY BCA(B5)**

**DIV:-B**

#include<iostream>

using namespace std;

int main()

{

int x=45;

int \*ptr1=&x;

int \*\*ptr2=&ptr1;

cout<<"\n value of x variable is = "<<x;

cout<<"\n value of singal pointer is ="<<\*ptr1;

cout<<"\n value of double pointer is ="<<\*\*ptr2;

return 0;

}

**Output:**

value of x variable is = 45

value of singal pointer is =45

value of double pointer is =45

**Practical:-14(b) Write a program to create pointers that point to objects and access their members.**

**Name:-Pranav Rajesh Tayade**

**Roll No:-153**

**Class:-FY BCA(B5)**

**DIV:-B**

#include<iostream>

using namespace std;

class Vehicle

{

public:

string type;

string maker;

int year;

void show() {

cout << "The vehicle type is a " << type

<< ", built by " << maker

<< " and Year is " << year << endl;

}

};

int main()

{

Vehicle veh;

Vehicle \*vehPtr = &veh;

vehPtr->type = "Car";

vehPtr->maker = "BMW";

vehPtr->year = 2025;

vehPtr->show();

return 0;

}

**Output:**

The vehicle type is a Car, built by BMW and Year is 2025

**Practical:-14(c) Write a program to demonstrate the use of pointers to functions.**

**Name:-Pranav Rajesh Tayade**

**Roll No:-153**

**Class:-FY BCA(B5)**

**DIV:-B**

#include <iostream>

using namespace std;

int multiply(int a, int b) { return a \* b; }

int main()

{

int (\*func)(int, int);

// func is pointing to the multiplyTwoValues function

func = multiply;

int prod = func(15, 2);

cout << "The value of the product is: " << prod << endl;

return 0;

}

**Output:**

The value of the product is: 30

**Practical:-15 Write a program to demonstrate use of Exception Handling.**

**Name:-Pranav Rajesh Tayade**

**Roll No:-153**

**Class:-FY BCA(B5)**

**DIV:-B**

#include <iostream>

using namespace std;

int main() {

int x,y;

float z;

cout<<"\nEnter two integears ";

cin >>x>>y;

try {

if (y == 0)

throw 0;

z=(float)x/y;

cout<<"\nDivision = "<<z;

}

catch (int num\_exception) {

cout <<"Error: Cannot divide by "<< num\_exception;

}

return 0;

}

**Output:**

Enter two integears 12 0

Error: Cannot divide by 0

**Practical:-16 (a)Write a program to find the largest and smallest elements from an array**

**Name:-Pranav Rajesh Tayade**

**Roll No:-153**

**Class:-FY BCA(B5)**

**DIV:-B**

#include<iostream>

using namespace std;

int main ()

{

int arr[10], n, i, max, min;

cout <<"Enter the size of the array : ";

cin >> n;

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 <<"\nLargest element : " << max;

cout <<"\nSmallest element : " << min;

return 0;

}

**Output:**

Enter the size of the array : 5

Enter the elements of the array : 23 4 5 766 34

Largest element : 766

Smallest element : 4

**Practical:-16 (b) Write a program to sort an array in ascending and descending order.**

**Name:-Pranav Rajesh Tayade**

**Roll No:-153**

**Class:-FY BCA(B5)**

**DIV:-B**

#include<iostream>

using namespace std;

void ascDecFunc(int a[], int n)

{

int temp;

for(int i=0;i < n-1;i++)

{

for(int j = 0;j < n/2; j++)

{

if(a[j]>a[j+1])

{

temp=a[j];

a[j]=a[j+1];

a[j+1]=temp;

}

}

for(int j = n/2;j < n-1; j++)

{

if(a[j] < a[j+1])

{

temp=a[j];

a[j]=a[j+1];

a[j+1]=temp;

}

}

}

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

cout<<a[i]<<" ";

}

int main()

{

int arr[] = {3, 2, 4, 1, 10, 30, 40, 20};

int len = sizeof(arr) / sizeof(arr[0]);

ascDecFunc(arr, len);

return 0;

}

**Output:**

1 2 3 4 40 30 20 10

**Practical:-17 Write a program to concatenate two strings and find the length of a string**

**Name:-Pranav Rajesh Tayade**

**Roll No:-153**

**Class:-FY BCA(B5)**

**DIV:-B**

#include<iostream>

#include<cstring>

using namespace std;

int main()

{

char str1[100], str2[100], result[200];

cout << "Enter the first string: ";

cin.getline(str1, 100);

cout << "Enter the second string: ";

cin.getline(str2, 100);

// Copy str1 to result

strcpy(result, str1);

// Concatenate str2 to result

strcat(result, str2);

// Output the result

cout << "Concatenated String: " << result << endl;

cout << "Length of Concatenated String: " << strlen(result) << endl;

return 0;

}

**Output:**

Enter the first string: pranav

Enter the second string: Tayade

Concatenated String: pranavTayade

Length of Concatenated String: 12

**Practical:-18 Write a program to calculate the factorial of a number using recursion**

**Name:-Pranav Rajesh Tayade**

**Roll No:-153**

**Class:-FY BCA(B5)**

**DIV:-B**

#include<iostream>

using namespace std;

int factorial(int n);

int main() {

int n;

cout << "Enter a positive integer: ";

cin >> n;

cout << "Factorial of " << n << " = " << factorial(n);

return 0;

}

int factorial(int n)

{

if(n > 1)

return n \* factorial(n - 1);

else

return 1;

}

**Output:**

Enter a positive integer: 5

Factorial of 5 = 120

**Practical:-19 Write a program that demonstrates different types of polymorphism (e.g.method overriding).**

**Name:-Pranav Rajesh Tayade**

**Roll No:-153**

**Class:-FY BCA(B5)**

**DIV:-B**

#include<iostream>

using namespace std;

class Animal {

public:

void eat(){

cout<<"Eating...";

}

};

class Dog: public Animal

{

public:

void eat()

{

cout<<"Eating bread...";

}

};

int main()

{

Dog d;

d.eat();

return 0;

}

**Output:**

Eating bread...

**Practical:-20 Write a program that demonstrates use of File handling**

**Name:-Pranav Rajesh Tayade**

**Roll No:-153**

**Class:-FY BCA(B5)**

**DIV:-B**

#include <iostream>

#include <fstream>

using namespace std;

int main() {

// Write to file

ofstream outFile("simple.txt");

outFile << "Hello, this is a simple file handling example.";

outFile.close();

// Read from file

ifstream inFile("simple.txt");

string content;

getline(inFile, content);

cout << "File content: " << content << endl;

inFile.close();

return 0;

}

**Output:**

File content: Hello, this is a simple file handling example.

**Practical:-21 Write a program to allocate and deallocate memory dynamically using**

**Name:-Pranav Rajesh Tayade**

**Roll No:-153**

**Class:-FY BCA(B5)**

**DIV:-B**

#include <iostream>

using namespace std;

int main()

{

int\* arr = new int[5];

cout << "Enter 5 values in the array: ";

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

{

cin >> arr[i];

}

cout << "\nArray elements: ";

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

{

cout << arr[i] << " ";

}

delete [] arr;

cout << "\nGarbage array values after deallocation of array memory: ";

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

{

cout << arr[i] << " ";

}

return 0;

}

**Output:**

Enter 5 values in the array: 1 2 3 4 5

Array elements: 1 2 3 4 5

Garbage array values after deallocation of array memory: 12392992 0 12386640 0 5