1.)Default constructor:

Write a class Student with a default constructor that initializes the student's name to "Unknown" and age to 0. Add a method display to print the student's details.

#include<iostream> //header file

Using namespace std;

Public:

String name;

Int age;

Student(): name(“Haritha”),age(23){} //default constructor

Void display()const{ //display the student details

Std::cout<<”name:”<<”\nage:”<<age<<std::endl;

}

};

Int main(){

Student student; //to create an instance of student using default constructor

Student.display(); //to display the student details

Return 0;

}

2.) 2. \*Parameterized Constructor:\*

Write a class Rectangle with a parameterized constructor that initializes the length and width. Add a method area that returns the area of the rectangle.

#include<iostream>

Class rectangle{

Public:

Double length;

Double width;

};

Void rectangle(double len, double wid):length(len),width(wid){} //parameterized constructo

Double area() const{ //to calculate area of rectangle

Return length\*width;

}

};

Int main()

Rectangle rect(8.0,3.0); //create a rectangle with given length and width

Std::cout<<”the area of the rectangle is :”<<rect.area()<<std::endl; //display the area of rect

Return0;

}

Pointers

Pointer to an integer:

Write a function increment that takes a pointer to an integer and increments its value by 1. Demonstrate the function in the main program.

#include<iostream>

Void increment(int \*num){

(\*num)++;

}

Int main(){ //main function

Int value=5; //take 5 as integer

Std::cout<<”initial value:”<<value<<std::endl; //original values

Increment(&value);

Std::cout<<”value after increment:”<<value<<std::endl; //to increment the val

Return 0; //print the result

}

14. \*Reference to an Integer:\*

Write a function swap that takes two integer references and swaps their values. Demonstrate the function in the main program.

#include<iostream>

Void swap(int& a,int& b)

{

Int temp=a; //swapping of two integers

a=b;

b=temp;

}

Int main()

{

Int x=5; //take x,y values as integers

Y=10;

Std::cout<<”before swapping:x=”<<x<<”,y=”<<y<<std::endl; //original values

Swap(x,y); //swap the values

Std::cout<<”after swapping:x=”<<x<<”,y=”<<y<<std::endl; //after swapping val

Return 0; //print the result

}

17. \*Pass by Value:\*

Write a function addTen that takes an integer by value and adds 10 to it. Demonstrate how the original value is not changed after calling the function.

#include<iostream> //header

Void addten(int num){ //add 10 as integer(pass by value)

Num += 10;

}

Int main() //main function

{

Int originalvalue=5;

Std::cout<<”original value:”<<original value<<std::endl;

Addten(original value); //call the addten function

Std::cout<<”value after calling addten:”<<original value<<std::endl;

Return 0; //print the result

}

15. \*Reference to a Class Object:\*

Write a class Box with a method volume. Create an object of this class and a reference to this object. Call the volume method using the reference.

#include <iostream>

class Box {

private:

double length;

double width;

double height;

public:

Box(double l, double w, double h) : length(l), width(w), height(h) {}

double volume() {

return length \* width \* height;

}

};

int main() {

Box myBox(3.0, 4.0, 5.0);

Box& boxRef = myBox;

std::cout << "Volume of the box: " << boxRef.volume() << std::endl;

return 0;

}

11. \*Array of Pointers:\*

Write a program that creates an array of pointers to integers. Initialize the array with values and print them using the pointers.

#include <iostream>

int main() {

const int SIZE = 5;

int\* ptrArray[SIZE];

int values[SIZE] = {10, 20, 30, 40, 50};

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

ptrArray[i] = &values[i];

}

std::cout << "Values using pointers:" << std::endl;

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

{

std::cout << \*ptrArray[i] << " ";

}

std::cout << std::endl;

return 0;

}

18. \*Pass by Reference:\*

Write a function addTenRef that takes an integer by reference and adds 10 to it. Demonstrate how the original value is changed after calling the function.

#include <iostream>

void addTenRef(int& num) {

num += 10;

}

int main() {

int originalValue = 5;

std::cout << "Original value: " << originalValue << std::endl;

addTenRef(originalValue);

std::cout << "Value after calling addTenRef: " << originalValue << std::endl;

return 0;

}