**OBJECT ORIENTED PROGRAMMING**

**LAB ASSIGNMENT -1**

**(Topics – Basics of C++, Classes and Objects)**

**Programming Questions**

1. **Write a program (WAP) to display "Hello World" on console display. WAP to implement the following control characters:**

**‘\n’ is for new line, or you can use *endl* – cout<<endl<<“message”;**

**‘\t’ is for tab ; ‘\a’ is an alarm sound; ‘\r’ is carriage return to go to the beginning of the current line**

#include<iostream>

using namespace std;

int main(){

cout<<"Hello World";

cout<<”Hello World”<<endl;

cout<<"Hello\n World";

cout<<"Hello\t World";

cout<<"Hello\a World";

cout<<"Hell\ro World";

return 0;

}

1. **Write a C++ program that will ask for a temperature in Celsius and display it in degree Fahrenheit.[F=9C/5+32]**

#include<iostream>

using namespace std;

int main(){

float f;

float c;

cout<<"Enter temperature in celsius: "<<endl;

cin>>c;

f=(9\*c)/5+32;

cout<<"The temperature in fahrenheit is: "<<f;

}

1. **WAP to demonstrate for, while, do-while (with all possible variations), like for loop can be demonstrated without giving initialization in for construct or without giving increment in for construct.**

**Sample:**

**for (int i=0; i<10; i++)**

**i=0**

**for (; i<10; i++)**

**i=0**

**for (; i<10;)**

**i++**

#include<iostream>

using namespace std;

int main(){

int num;

int i;

cout<<"Enter number: "<<endl;

cin>>num;

cout<<"Numbers less than entered number are: ";

//using for loop

for(i=1;i<=num;i++){

cout<<" "<<i;

}

//using while loop

cout<<"\nNumbers less than entered number are: ";

i=1;

while(i<=num){

cout<<" "<<i;

i++;

}

//using dowhile loop

i=1;

cout<<"\nNumbers less than entered number are: ";

do{

cout<<" "<<i;

i++;

}while(i<=num);

return 0;

}

1. **Create a structure in C++ containing the details of Students as details below and a main function to execute the structure.**

***Data Members(properties):***

***Name***

***Roll No***

***Degree***

***Hostel***

***CurrentCGPA***

***Member Function(behavior):***

***addDetails();***

***updateDetails();***

***updateCGPA();***

***updateHostel();***

***displaydetails();***

#include<iostream>

using namespace std;

struct Student{

char name[20];

int roll\_no;

char degree[10];

char hostel;

float currentcgpa;

void addDetails();

void updateDetails();

void updateCGPA();

void updateHostel();

void displaydetails();

};

void Student::addDetails(){

cout<<"Enter name"<<endl;

cin>>name;

cout<<"Enter roll\_no"<<endl;

cin>>roll\_no;

cout<<"Enter degree"<<endl;

cin>>degree;

cout<<"Enter hostel"<<endl;

cin>>hostel;

cout<<"Enter currentcgpa"<<endl;

cin>>currentcgpa;

}

void Student::displaydetails(){

cout<<"Name: "<<name<<endl;

cout<<"Roll\_no: "<<roll\_no<<endl;

cout<<"Degree: "<<degree<<endl;

cout<<"Hostel: "<<hostel<<endl;

cout<<"Current CGPA: "<<currentcgpa<<endl;

}

void Student::updateCGPA(){

int ans;

float newcgpa;

cout<<"Enter 1 to update CGPA\nEnter 0 to not update\nEnter answer: ";

cin>>ans;

if(ans==1){

cout<<"Enter updated CGPA: ";

cin>>currentcgpa;

}

else exit;

}

void Student::updateHostel(){

int ans;

float newcgpa;

cout<<"Enter 1 to update hostel\nEnter 0 to not update\nEnter answer: ";

cin>>ans;

if(ans==1){

cout<<"Enter updated hostel: ";

cin>>hostel;

}

else exit;

}

void Student::updateDetails(){

int ans;

float newcgpa;

cout<<"Enter 1 to update details\nEnter 0 to not update\nEnter answer: ";

cin>>ans;

if(ans==1){

cout<<"Enter updated name: ";

cin>>name;

cout<<"Enter updated roll number: ";

cin>>roll\_no;

cout<<"Enter updated degree: ";

cin>>degree;

}

else exit;

}

int main(){

Student S1;

S1.addDetails();

S1.updateCGPA();

S1.updateHostel();

S1.updateDetails();

S1.displaydetails();

return 0;

}

1. **Differentiate between private and public access/scope. Perform the question no. 4 with class instead of structure with having the data members private and some member functions in private scope and some in public scope.**

#include<iostream>

using namespace std;

class Student{

private:

char name[20];

int roll\_no;

char degree[10];

char hostel;

float currentcgpa;

void updateCGPA();

void updateHostel();

public:

void addDetails();

void updateDetails();

void displaydetails();

};

void Student::addDetails(){

cout<<"Enter name"<<endl;

cin>>name;

cout<<"Enter roll\_no"<<endl;

cin>>roll\_no;

cout<<"Enter degree"<<endl;

cin>>degree;

cout<<"Enter hostel"<<endl;

cin>>hostel;

cout<<"Enter currentcgpa"<<endl;

cin>>currentcgpa;

}

void Student::displaydetails(){

cout<<"Name: "<<name<<endl;

cout<<"Roll\_no: "<<roll\_no<<endl;

cout<<"Degree: "<<degree<<endl;

cout<<"Hostel: "<<hostel<<endl;

cout<<"Current CGPA: "<<currentcgpa<<endl;

}

void Student::updateCGPA(){

int ans;

float newcgpa;

cout<<"Enter 1 to update CGPA\nEnter 0 to not update\nEnter answer: ";

cin>>ans;

if(ans==1){

cout<<"Enter updated CGPA: ";

cin>>currentcgpa;

}

else exit;

}

void Student::updateHostel(){

int ans;

float newcgpa;

cout<<"Enter 1 to update hostel\nEnter 0 to not update\nEnter answer: ";

cin>>ans;

if(ans==1){

cout<<"Enter updated hostel: ";

cin>>hostel;

}

else exit;

}

void Student::updateDetails(){

int ans;

float newcgpa;

cout<<"Enter 1 to update details\nEnter 0 to not update\nEnter answer: ";

cin>>ans;

if(ans==1){

cout<<"Enter updated name: ";

cin>>name;

cout<<"Enter updated roll number: ";

cin>>roll\_no;

cout<<"Enter updated degree: ";

cin>>degree;

}

else exit;

updateCGPA();

updateHostel();

}

int main(){

Student S1;

S1.addDetails();

S1.updateDetails();

S1.displaydetails();

return 0;

}

1. **Create a code snippet that illustrates the following:**
   1. **Calling of private member functions inside public member function**
   2. **Access private member functions inside public member function**

#include<iostream>

using namespace std;

class test{

private:

int num1;

int num2;

int sum(int,int);

public:

int getnum();

int display();

};

int test::getnum(){

cout<<"Enter numbers: ";

cin>>num1>>num2;

}

int test::display(){

cout<<"Sum is: "<<test::sum(num1,num2);

}

int test::sum(int num1,int num2){

return num1+num2;

}

int main(){

test obj;

obj.getnum();

obj.display();

return 0;

}

1. **Define a class named *Complex* with properties (real and imaginary) and methods as per following details.**

***void set ()* to initialize object values.**

***void display ()* to display complex number.**

***Complex sum (Complex*) or *void sum (Complex)* to add two complex numbers (objects of Complex class)and *return complex\_number* (object of Complex class) as result*.***

**Properties (real and imaginary) of the code should have private access modifier and member functions should have public access modifier in C++ class.**

#include<iostream>

using namespace std;

class Complex{

private:

float real;

float img;

public:

void set(void);

void display(void);

Complex sum(Complex);

}C1,C2,C3;

void Complex::set(void){

cout<<"Enter real and imaginary part of number: ";

cin>>real>>img;

}

void Complex::display(){

cout<<real<<" + "<<img<<"i"<<endl;

}

Complex Complex::sum(Complex C3){

C3.real=C1.real+C2.real;

C3.img=C1.img+C2.img;

return C3;

}

int main(){

C1.set();

C1.display();

C2.set();

C2.display();

C3=C3.sum(C3);

C3.display();

return 0;

}

1. **Implement *namespace* in a program to illustrate the use of same name variables and functions in different sections/libraries of the code.**

#include<iostream>

using namespace std;

namespace one{

int i=1;

}

namespace two{

int i(int num1,int num2){

return num1\*num2;

}

}

using namespace one;

using namespace two;

int main(){

int num1;

int num2;

cout<<"i is: "<<one::i<<endl;

cout<<"Enter 2 numbers: ";

cin>>num1>>num2;

cout<<"Product is: "<<two::i(num1,num2);

return 0;

}