**Sukuna Multiple Campus**

Name: Ajay Kumar Sah

Symbol no: 76214001

Subject: OOP with C++

Submitted To: Uma Dungel

// function overloading

#include <iostream>

#include <conio.h>

using namespace std;

class Func\_overloading{

public:

void print(int a){

cout<<"inter is "<<a<<endl;

}

void print(double a)

{

cout<<"Double is "<<a<<endl;

}

void print(string a){

cout<<"Character is "<<a<<endl;

}

};

int main()

{

Func\_overloading f1;

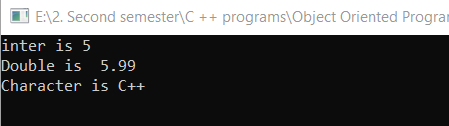
f1.print(5);

f1.print(5.99);

f1.print("C++");

return 0;

}



//inline Function

#include <iostream>

#include <conio.h>

using namespace std;

class Student

{

private:

int roll;

char name[25];

public:

void display();

void getdata();

};

inline void Student::getdata(){

cout<<"Enter student name: ";

cin>>name;

cout<<"Enter roll no ";

cin>>roll;

}

inline void Student::display(){

cout<<endl<<"Displaying the data of the student "<<endl<<endl;

cout<<"Student name: "<<name;

cout<<endl<<"Roll no: "<<roll;

}

int main(){

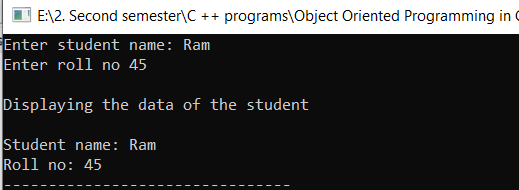
Student s1;

s1.getdata();

s1.display();

return 0;

}



//Encapsulation

#include <iostream>

#include <conio.h>

using namespace std;

class Encaps

{

private:

int a,b;

public:

void set\_data()

{

cout<<"Enter two numbers ";

cin>>a>>b;

}

void getdata()

{

cout<<"Addition of two number is "<<a+b;

}

};

int main()

{

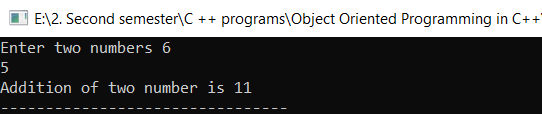
Encaps E1;

E1.set\_data();

E1.getdata();

return 0;

}



// Default Constructor

#include <iostream>

#include <conio.h>

using namespace std;

class DefaultConstructor{

public:

DefaultConstructor(){

cout<<"This is an example of default constructor\n";

}

};

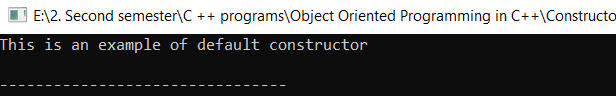
int main()

{

DefaultConstructor d1;

return 0;

}



//template class

#include <iostream>

#include <conio.h>

using namespace std;

template <class temp>

class Calculator

{

temp n1,n2;

public:

Calculator(temp n1,temp n2)

{

this->n1=n1;//this->n1 is above private access specifier member

this->n2=n2;

}

void display()

{

cout<<"NUmber are "<<n1<<" "<<n2;

cout<<endl<<"Addition "<<add()<<endl;

}

temp add()

{

return n1+n2;

}

};

int main()

{

Calculator <int> cal(5,10);

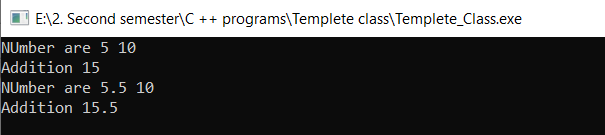
Calculator <float> f(5.5,10);

cal.display();

f.display();

return 0;

}



//Single level inheritance

#include <iostream>

#include <conio.h>

using namespace std;

class Base\_class{

protected:

int age;

};

class Child\_class: public Base\_class{

public:

void myage()

{

age=18;

cout<<"your age is "<<age;

}

};

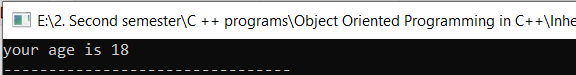
int main(){

Child\_class ch;

ch.myage();

return 0;

}



//Array of object

#include <iostream>

#include <conio.h>

using namespace std;

class largest{

private:

int a,b;

public:

void getdata();

int largestdata();

void displaydata();

};

void largest:: getdata(){

cout<<"Enter numbers ";

cin>>a>>b;

}

int largest::largestdata(){

//returns one of the largest value among two

if(a>=b){

return a;

}

else{

return b;

}

}

void largest::displaydata(){

if(largestdata()>=largestdata())

{

cout<<endl<<"The largest value is "<<largestdata()<<endl;

}

}

int main(){

int i;

largest l1[2];

for(i=0;i<2;i++){

l1[i].getdata();

}

for(i=0;i<2;i++){

l1[i].largestdata();

l1[i].displaydata();

}

return 0 ;

}

