**Assignment 1:**

#include<iostream>

using namespace std;

class bank{

public:

int acc\_no,bal,deposit,withdraw;

char name[20];

char acc\_type[20];

public:

void detail(){

cout<<"Enter your details :";

cout<<endl<<endl;

cout<<"NAME :\n";

cin>>name;

cout<<"ACCONT NO :\n";

cin>>acc\_no;

cout<<"ACCONT TYPE :\n";

cin>>acc\_type;

cout<<"Enter opening balance: ";

cin>>bal;

}

void deposit1(){

cout<<"Enter amount to deposit : \n";

cin>>deposit;

bal=bal+deposit;

cout<<"Total balance after deposition : \n";

cout<<bal;

}

void withdraw1(){

cout<<"Enter amount to withdraw : \n";

cin>>withdraw;

bal=bal-withdraw;

cout<<"Total balance after withdrawing the amount : \n";

cout<<bal;

}

void display(){

cout<<"Your details are :"<<endl<<endl;

cout<<"NAME :"<<name<<endl;

cout<<"ACCOUNT NO :"<<acc\_no<<endl;

cout<<"ACCOUNT TYPE :"<<acc\_type<<endl;

cout<<"TOTAL BALANCE :"<<bal<<endl;

}

};

int main(){

int i=0;

int choice;

do{

cout<<"\nCHOICE FROM BELOW:"<<endl<<endl;

cout<<"1.Enter your details."<<endl;

cout<<"2.Deposit amount."<<endl;

cout<<"3.Withdraw amount."<<endl;

cout<<"4.Display your details."<<endl;

cout<<"5.Exit.";

cin>>choice;

i++;

bank b;

switch(choice){

case 1: b.detail();

break;

case 2: b.deposit1();

break;

case 3: b.withdraw1();

break;

case 4: b.display();

break;

}

}while(i<5);

return 0;

}

**Assignment 2:**

#include<iostream>

using namespace std;

class vehicles

{

    public:

    int mileage ,price;

    };

    class car : public vehicles

    {

        public:

        int owernership\_cost, warrenty,capacity;

        char fuel\_type[20];

    };

    class audi : public car

    {

      char model\_type[20];

    };

     class Ford : public car

    {

      char model\_type[20];

    };

     class bike : public vehicles

    {

        public:

        int no\_cylinder, no\_gears;

        char cooling\_type[20];

    };

     class bajaj : public bike

    {

      char make\_type[20];

    };

     class tvs : public bike

    {

      char make\_type[20];

    };

int main()

{

    audi a;

    cout<<"ENTER DETAILS OF AUDI :\n";

    cout<<"Enter Owernership cost:";

    cin>>a.owernership\_cost;

    cout<<"Enter warrenty of car:";

    cin>>a.warrenty;

    cout<<"Enter capacity:";

    cin>>a.capacity;

    cout<<"Enter fuel type:";

    cin>>a.fuel\_type;

    cout<<"Enter mileage:";

    cin>>a.mileage;

    cout<<"Enter price:";

    cin>>a.price;

    cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

    Ford f;

    cout<<"ENTER DETAILS OF FORD :\n";

    cout<<"Enter Owernership cost:";

    cin>>f.owernership\_cost;

    cout<<"Enter warrenty of car:";

    cin>>f.warrenty;

    cout<<"Enter capacity:";

    cin>>f.capacity;

    cout<<"Enter fuel type:";

    cin>>f.fuel\_type;

    cout<<"Enter mileage:";

    cin>>f.mileage;

    cout<<"Enter price:";

    cin>>f.price;

int choice;

cout<<"Choose the car :\n";

cout<<"1. AUDI\n";

cout<<"2. FORD\n";

cin>>choice;

switch(choice)

{

    case 1:

    cout<<"AUDI INFO : \n";

    cout<<"Cost : "<<a.owernership\_cost<<"Rupees\n";

    cout<<" Warrenty : "<<a.warrenty<<"Years\n";

     cout<<"Capacity : "<<a.capacity<<"Seats\n";

     cout<<"Fuel Type: "<<a.fuel\_type<<"\n";

     cout<<"Mileage: "<<a.mileage<<"Kmpl\n";

    cout<<"Price: "<<a.price<<"Rupees\n";

    break;

     case 2:

    cout<<"FORD INFO : \n";

    cout<<"Cost : "<<f.owernership\_cost<<"Rupees\n";

    cout<<"Warrenty : "<<f.warrenty<<" Years\n";

    cout<<"Capacity : "<<f.capacity<<"Seats\n";

    cout<<"Fuel Type: "<<f.fuel\_type<<"\n";

    cout<<"Mileage: "<<f.mileage<<"Kmpl\n";

    cout<<"Price: "<<f.price<<"Rupees\n";

    break;

}

    return 0;

}

**Assignment 3:**

#include<iostream>

using namespace std;

class shape

{

    public:

    double l,b;

    void getdata()

    {

        cout<<"Enter Length: ";

        cin>>l;

        cout<<"Enter Breadth: ";

        cin>>b;

    }

    virtual void display\_area()=0;  //USING PURE VIRTUAL FUNCTION

};

class triangle:public shape

{

    public:

    void display\_area()

    {

        double area=0.5\*l\*b;

        cout<<"Area of Triangle: "<<area<<endl;

    }

};

class rectangle:public shape

{

    public:

    void display\_area()

    {

        double area=l\*b;

        cout<<"Area of Rectangle: "<<area;

    }

};

int main()

{

    int choice;

    for(int i=0;i<2;i++)

    {

        cout<<"\n \*\*\*\*\*\*\*\*\*\*CHOOSE A SHAPE\*\*\*\*\*\*\*\*";

        cout<<"\n 1. TRIANGLE";

        cout<<"\n 2. RECTANGLE\n";

        cin>>choice;

        if(choice==1)

        {

            //triangle t;

            //t.getdata();

            //t.display\_area();

            triangle t;

            shape \*sh=&t;

            sh->getdata();

            sh->display\_area();

        }

        else if(choice==2)

        {

            //rectangle r;

            //r.getdata();

            //r.display\_area();

            rectangle r;

            shape \*sh=&r;

            sh->getdata();

            sh->display\_area();

        }

        else

        {

            cout<<"INVALID CHOICE";

        }

    }

    return 0;

}

**Assignment 4:**

#include<iostream>

using namespace std;

class complex

{

    float x;

    float y;

    public:

    complex()

    {

    x=0;

    y=0;

    }

    complex operator+(complex);

    complex operator\*(complex);

    friend istream & operator >>(istream & input,complex & t)

    {

        cout<<"Enter the real part :";

        input>>t.x;

        cout<<"Enter the imaginary part : ";

        input>>t.y;

    }

    friend ostream &operator<<(ostream & output, complex & t)

    {

        output<<t.x<<"+"<<t.y<<"i\n";

    }

};

complex complex::operator+(complex c)

{

    complex temp;

    temp.x=x+c.x;

    temp.y=y+c.y;

    return(temp);

}

complex complex::operator\*(complex c)

{

    complex temp2;

    temp2.x=(x\*c.x)-(y\*c.y);

    temp2.y=(y\*c.x)+(x\*c.y);

    return(temp2);

}

int main()

{

    complex c1,c2,c3,c4;

    cout<<"Default constructor value : \n";

    cout<<c1;

    cout<<"Enter the 1st number\n";

    cin>>c1;

    cout<<"Enter the 2nd number\n";

    cin>>c2;

    c3=c1+c2;

    c4=c1\*c2;

    cout<<"\n The first number is :";

    cout<<c1;

    cout<<"\n The second number is :";

    cout<<c2;

    cout<<"\n Addition :";

    cout<<c3;

    cout<<"\n Multiplication :";

    cout<<c4;

    return 0;

}

**Assignment 5:**

#include<iostream>

using namespace std;

class matrix

{

    public:

    int a,b,c,d;

    int m1[10][10];

    int m2[10][10];

    int m3[10][10];

    void read()

    {

        cout<<"Enter number of rows for matix 1 :";

        cin>>a;

        cout<<"Enter number of cols for matix 1:";

        cin>>b;

        cout<<"Enter number of rows for matix 2:";

        cin>>c;

        cout<<"Enter number of cols for matix 2:";

        cin>>d;

        try

        {

            if((a>10)||(b>10))

            {

                throw 1;

            }

        }

        catch(int)

        {

            cout<<"No. of rows and columns can't be more than 10!!";

            return;

        }

        try

        {

            if(c>10 ||d>10)

            {

                throw 1;

            }

        }

        catch(int)

        {

            cout<<"No. of rows and columns can't be more than 10!!";

        }

        cout<<"Enter values of Matrix 1 :"<<endl;

        for(int i=0;i<a;i++)

        {

            for(int j=0;j<b;j++)

            {

                cin>>m1[i][j];

            }

        }

        cout<<"Enter values of Matrix 2 :"<<endl;

        for(int i=0;i<a;i++)

        {

            for(int j=0;j<b;j++)

            {

                cin>>m2[i][j];

            }

        }

        if((a==c)&&(b==d)){

            cout<<"\nAddition of Matrices is :"<<endl;

            for( int i=0;i<a;i++)

            {

                for(int j=0;j<b;j++)

                {

                    cout<<m1[i][j]+m2[i][j]<<" ";

                }

                cout<<endl;

            }

            cout<<"\nSubstraction of Matrices is :"<<endl;

            for( int i=0;i<a;i++)

            {

                for(int j=0;j<b;j++)

                {

                    cout<<m1[i][j]-m2[i][j]<<" ";

                }

                cout<<endl;

            }

        }

        else

        {

            try

            {

                throw 1;

            }

            catch(int)

            {

                cout<<"Due to order mismatch couldn't perform addtion and substraction!!";

            }

        }

        if(b==c)

        {

            for( int i=0;i<a;i++)

            {

                for(int j=0;j<b;j++)

                {

                    m3[i][j]=0;

                    for(int k=0;k<c;k++)

                    {

                        m3[i][j]+= m1[i][k] \* m2[k][j];

                    }

                }

                cout<<endl;

            }

            cout<<"\nMultiplication of Matrices is :"<<endl;

            for(int i=0;i<a;i++)

            {

                for(int j=0;j<b;j++)

                {

                    cout<<m3[i][j]<<" ";

                }

                cout<<endl;

            }

        }

        else

        {

            try

            {

                throw 1;

            }

            catch(int)

            {

                cout<<"Due to order mismatch couldn't perform multiplication!!";

            }

        }

}

};

int main()

{

    matrix m;

    m.read();

    return 0;

}

**Assignment 6:**

#include<iostream>

using namespace std;

class Student{

    public:

    int rollno;

    char s[30],name[90],s1[20];

    virtual void get(){

        cout<<"Enter Student Name: ";

        cin>>name;

        cout<<"Enetr the Roll no: ";

        cin>>rollno;

        cout<<"Enter the subject name: ";

        cin>>s1;

    }

};

class record:public Student{

    public:

    int i,u;

    void get()

    {

        Student::get();

        cout<<"Enetr the Subject code: ";

        cin>>s;

        cout<<"Enter the internal marks: ";

        cin>>i;

        cout<<"Enter the univercity marks: ";

        cin>>u;

    }

    void display()

    {

        cout<<"Student Name \t\t Roll No \t\t Subject \t\t Subject Code \t\t Internal Marks \t\t University marks \n";

        cout<<name<<"\t\t"<<rollno<<"\t\t"<<s1<<"\t\t"<<s<<"\t\t"<<i<<"\t\t"<<u<<"\n";

    }

};

int main()

{

    Student \*stu;

    record r;

    stu=&r;

    stu->get();

    r.display();

    return 0;

}

**Assignment 7:**

#include<iostream>

using namespace std;

class Except

{

    public:

    int i,array[5],ele;

    float num1,num2;

    void Take(){

        cout<<"Eneter Array Element: "<<endl;

        for(i=0;i<5;i++)

        {

            cin>>array[i];

        }

        cout<<"Enter Num1: ";

        cin>>num1;

        cout<<"Enter Num2: ";

        cin>>num2;

    }

    void Exception1()

    {

        cout<<"\*\*\*Access Array Element\*\*\*"<<endl;

        cout<<"Enter Element to access: "<<endl;

        cin>>i;

        try

        {

            if(i<5)

            {

                cout<<"Array element present at "<<i<<" is "<<array[i];

            }

            else

            {

                throw 1;

            }

        }

        catch(int)

        {

            cout<<"ArrayIndexedBoundException !!"<<endl;

        }

    }

    void Exception2()

    {

        try

        {

            if(num2==0)

            {

                throw 1;

            }

        }

        catch(int)

            {

                cout<<"NumberFormatException !!"<<endl;

            }

            try

            {

                if(num1!=(int)num1 && num2!=(int)num2)

                {

                    cout<<"Num1/num2= "<<num1/num2<<endl;

                }

                else

                {

                    throw 2;

                }

            }

            catch(int)

            {

                cout<<"NumberFormatException !!"<<endl;

            }

        }

};

int main()

{

    Except ex;

    ex.Take();

    ex.Exception1();

    ex.Exception2();

    return 0;

}

**Assignment 9:**

//write c++ program using STL map for mapping person record (name, telephone no ) perform operations - add, display, search, delete, update.

#include <iostream>

#include <string>

#include <cstdlib>

#include <map>

using namespace std;

int main(){

    map<int, string> mp;

    map<int, string>::iterator itr;

    int choice,tel\_no,n;

    string name,new\_name;

    while(true)

    {

    cout<<"Map implementation in STL\n";

    cout<<"1. Insert Element into the Map.\n";

    cout<<"2. Display the Map Elements.\n";

    cout<<"3. Search Element at a Key in Map.\n";

    cout<<"4. Delete Element of the Map.\n";

    cout<<"5. Update the value of given key:\n";

    cout<<"6. Exit\n\n";

    cout<<"Enter your choice :\n";

    cin>>choice;

    switch(choice)

    {

    case 1:

    cout<<"Enter the key :\n";

    cin>>tel\_no;

    cout<<"Enter the value to be inserted :\n";

    cin>>name;

    mp.insert(pair<int, string>(tel\_no, name));

    break;

    case 2:

    for( itr = mp.begin(); itr != mp.end(); ++itr)

    {

    cout << "Key => " << itr->first << ", Value => " << itr->second << endl;

    }

    break;

    case 3:

    cout<<"Enter the key at which value to be found :\n";

    cin>>tel\_no;

    cout<<mp.find(tel\_no)->second<<endl;

    break;

    case 4:

    cout<<"Enter the mapped string to be deleted: ";

    cin>>tel\_no;

    mp.erase(tel\_no);

    break;

    case 5:

    cout<<"Enter the key whose value you want to update :";

    cin>>n;

    itr = mp.find(n);

    if (itr != mp.end())

    cout<<"Enter new value :";

    cin>>new\_name;

    itr->second = new\_name;

    cout<<"Map after updation :\n";

    for( itr = mp.begin(); itr != mp.end(); ++itr)

    {

    cout << "Key => " << itr->first << ", Value => " << itr->second << endl;

    }

    break;

    case 6:

    exit(1);

    break;

    default:

    cout<<"Wrong Choice!!";

    break;

    }

    }

    return 0;

}

**Assignment 10:**

import java.util.Scanner;

public class Assignment10

{

    pubilc static void main(String args[]){

        Scanner sc=new Scanner(System.in);

        System.out.println("Enter basic salary: ");

        Double basic=sc.nextDouble();

        double hra=0.10\*basic;

        double Ta=0.05\*basic;

        double grosssalary=basic+hra+Ta;

        double tax=0.02\*grosssalary;

        double netSalary=grosssalary-tax;

        System.out.println("\*\*\*\*\*\*\*\*\*Employee salary breackdown\*\*\*\*\*\*\*\*\*");

        System.out.println("Basic Pay: "+basic);

        System.out.println("HRA: "+hra);

        System.out.println("TA: "+Ta);

        System.out.println("Gross Salary: "+grosssalary);

        System.out.println("Tax: "+tax);

        System.out.println("Net Salary: "+netSalary);

    }

}

**Assignment 11:**

import java.util.Scanner;

public class Assignment11

{

    public static void main(String[] args)

    {

        int num,digit,temp,cubesum=0;

        System.out.println("Enter Number: ");

        Scanner sc=new Scanner(System.in);

        num=sc.nextInt();

        temp=num;

        while(num!=0)

        {

            digit=num%10;

            cubesum+=Math.pow(digit,3);

            num/=10;

        }

        if(cubesum==temp)

        {

            System.out.println(temp+" is Armstrong!!");

        }

        else

        {

            System.out.println(temp+" is not an Armstrong!!");

        }

    }

}

**Write a Program to Implement a Class STUDENT having Following Members: Data members & Member functions, Accept Name of the student, marks of the student to Compute Total, Average to Display the Data.**

#include<iostream>

using namespace std;

class Student

{

    public:

    int s1,s2,s3;

    char name[50];

    void getdata()

    {

        cout<<"Student Name: ";

        cin>>name;

        cout<<"\nSubject1 Marks: ";

        cin>>s1;

        cout<<"\nSubject2 Marks: ";

        cin>>s2;

        cout<<"\nSubject3 Marks: ";

        cin>>s3;

    }

    void compute()

    {

        int total,avg;

        total = s1 + s2 + s3;

        avg= total / 3;

        cout<<"\nTotal: "<<total;

        cout<<"\nAverage: "<<avg;

    }

};

int main()

{

    Student obj;

    obj.getdata();

    obj.compute();

    return 0;

}

**Design a C++ Class ‘Complex‘ with data members for real and imaginary part. Provide default and parameterized constructors. Write a program to perform arithmetic operations of two complex numbers using operator overloading.**

#include<iostream>

using namespace std;

class complex

{

    float x;

    float y;

    public:

    complex()

    {

        x=0;

        y=0;

    }

    complex(float m,float n)

    {

        x=m;

        y=n;

    }

    void display()

    {

        cout<<x<<"+"<<y<<"i\n";

    }

    void operator+(complex &c)

    {

        x=x+c.x;

        y=y+c.y;

    }

    void operator\*(complex &c)

    {

        x=(x\*c.x)-(y\*c.y);

        y=(y\*c.x)+(x\*c.y);

    }

};

int main(){

    complex c1;

    cout<<"Default Constructor Value =";

    c1.display();

    complex c2(10,20);

    complex c3(11,12);

    cout<<"Addition is : ";

    c2+c3;

    c2.display();

    cout<<"Multiplication is : ";

    c2\*c3;

    c2.display();

    return 0;

}

**C++program to demonstrate example of Hierarchical inheritance to get square and cube of a number.**

#include<iostream>

using namespace std;

class Number

{

    public:

    int num;

    void getnum()

    {

        cout<<"\nEnter Number: ";

        cin>>num;

    }

};

class Square:public Number

{

    public:

    int square;

    void getsquare()

    {

        square = num \* 2;

        cout<<"\nSuare of "<<num<<": "<<square;

    }

};

class Cube:public Number

{

    public:

    int cube;

    void getCube()

    {

        cube = num \* 3;

        cout<<"\nCube of "<<num<<": "<<cube;

    }

};

int main()

{

    Square sq;

    Cube cu;

    sq.getnum();

    sq.getsquare();

    cu.getnum();

    cu.getCube();

    return 0;

}

**Implement C++ program to read and print employee information using multiple inheritance.**

#include <iostream>

using namespace std;

class Employee {

public:

  string name;

  int id;

  string department;

  Employee(string name, int id, string department) {

    this->name = name;

    this->id = id;

    this->department = department;

  }

  void printInfo() {

    cout << "Name: " << name << endl;

    cout << "Id: " << id << endl;

    cout << "Department: " << department << endl;

  }

};

class Salary {

    public:

        int salary;

        Salary(int salary) {

            this->salary = salary;

        }

        void printSalary() {

            cout << "Salary: " << salary << endl;

        }

};

class EmployeeInfo : public Employee, public Salary {

public:

  EmployeeInfo(string name, int id, string department, int salary) : Employee(name, id, department), Salary(salary) {}

  void printAllInfo() {

    printInfo();

    printSalary();

  }

};

int main() {

  EmployeeInfo employeeInfo("John Doe", 1234, "Sales", 50000);

  employeeInfo.printAllInfo();

  return 0;

}

**A book shop maintains the inventory of books that are being sold at the shop. The list includes details such as author, title, price, publisher and stock position. Whenever a customer wants a book, the sales person inputs the title and author and the system searches the list and displays whether it is available or not. If it is not, an appropriate message is displayed. If it is, then the system displays the book details and requests for the number of copies required. If the requested copies book details and requests for the number of copies required. If the requested copies are available, the total cost of the requested copies is displayed; otherwise the message “Required copies not in stock” is displayed.**

**Imagine a publishing company which does marketing for book and audio cassette versions. Create a class publication that stores the title (a string) and price (type float) of publications. From this class derive two classes: book which adds a page count (type int) and tape which adds a playing time in minutes (type float). Write a program that instantiates the book and tape class, allows user to enter data and displays the data members. If an exception is caught, replace all the data member values with zero values.**

**Implement C++ program to implement a base class consisting of the data members such as name of the student, roll number and subject. The derived class consists of the data members subject code, internal assessment and university examination marks. The program should have the facilities. i) Build a master table ii) List a table iii) Insert a new entry iv) Delete old entry v) Edit an entry vi) Search for a record. Use virtual functions.**

**Develop an object oriented program in C++ to create a database of the personnel information system containing the following information: Name, Date of Birth, Blood group, Height, Weight, Insurance Policy number, Contact address, telephone number, driving licence no. etc Construct the database with suitable member functions for initializing and destroying the data viz constructor, default constructor, copy constructor, destructor.**

[**https://sppusolutions.blogspot.com/2019/10/develop-object-oriented-program-in-c-to.html**](https://sppusolutions.blogspot.com/2019/10/develop-object-oriented-program-in-c-to.html)