

eBox Solutions

Classes and Objects

Problem 1: Classes and Objects II

Main.cpp

```
#include <iostream>
#include <string>

using namespace std;

class Donor
{
public:
    string name;
    int age;
    float height;
    float weight;
    string gender;
    string bloodGroup;

    void display()
    {
        cout<<"\nDonor details are :";
        cout<<"\nName :"<< name;
        cout<<"\nAge :"<<age;
        cout<<"\nheight :"<<height;
        cout<<"\nweight :"<<weight;
        cout<<"\nGender :"<<gender;
        cout<<"\nBlood Group :"<<bloodGroup;
    }
};

int main() {
    Donor d1;
    cout<<"\nEnter the donor details";
    cout<<"\nEnter the name :";
    cin>>d1.name;
    cout<<"\nEnter the age :";
    cin>>d1.age;
    cout<<"\nEnter the height :";
    cin>>d1.height;
    cout<<"\nEnter the weight :";
    cin>>d1.weight;
    cout<<"\nEnter the gender :";
    cin>>d1.gender;
    cout<<"\nEnter the blood group :";
    cin>>d1.bloodGroup;
    d1.display();
}
```

```
}
```

Problem 2: GETTERS AND SETTERS

Main.cpp

```
#include <iostream>
#include<string>
#include<stdio.h>
using namespace std;

class ItemType
{
    private:
        string name;
        double deposit;
        double costPerDay;
    public:
        void setName(string a){
            name = a;
        }
        void setDeposit(double d)
        {
            deposit = d;
        }
        void setCostPerDay(double d)
        {
            costPerDay = d;
        }
        void display()
        {
            cout<<"Itemtype details:";
            cout<<"\nName: "<<name;
            cout<<"\nDeposit: "<<deposit;
            cout<<"\nCost Per Day: "<<costPerDay;
        }
};

int main()
{
    ItemType i;
    string a;
    double b;
    cout<<"Enter name:"<<endl;
    getline(cin, a);
    i.setName(a);
    cout<<"Enter deposit:"<<endl;
    cin>>b;
    i.setDeposit(b);
    cout<<"Enter cost Per Day:"<<endl;
    cin>>b;
    i.setCostPerDay(b);
    i.display();
}
```

Problem 3: CLASSES AND OBJECTS IV

Main.cpp

```
#include <iostream>
#include <string>
#include <stdio.h>
#include <cstdlib>
using namespace std;
class Donor
{
public:
    string name;
    int age;
    float height, weight;
    string gender, bloodGroup;

    void display()
    {
        cout<<"\nName : "<<name;
        cout<<"\nAge : "<<age;
        cout<<"\nheight : "<<height;
        cout<<"\nweight : "<<weight;
        cout<<"\nGender : "<<gender;
        cout<<"\nBlood Group : "<<bloodGroup;
    }
};

int main()
{
    int d;
    cout<<"\nEnter the number of donors:";
    cin>>d;
    Donor to[10];
    for(int i=0; i<d; i++)
    {
        cout<<"\nEnter the donor details"<<i+1;
        cout<<"\nEnter the Name :";
        cin>>to[i].name;
        cout<<"\nEnter the Age :";
        cin>>to[i].age;
        cout<<"\nEnter the height :";
        cin>>to[i].height;
        cout<<"\nEnter the weight :";
        cin>>to[i].weight;
        cout<<"\nEnter the Gender :";
        cin>>to[i].gender;
        cout<<"\nEnter the Blood Group :";
        cin>>to[i].bloodGroup;
    }
    cout<<"\nDonor details in the hospital database :\n";
    for (int i=0;i<d;i++)
    {
        cout<<"Donor"<<i+1;
        to[i].display();
    }
}
```

```
}  
}
```

Problem 4: SUPERMARKET

Main.cpp

```
#include <iostream>  
#include<string>  
using namespace std;  
  
class Purchase  
{  
    private:  
        string userName;  
        int countOfItems;  
        float amount;  
        float static totalAmt;  
        int static totalCountOfItems;  
  
    public:  
  
        string getUserName()  
        {  
            return userName;  
        }  
  
        void setName(string a)  
        {  
            this->userName=a;  
        }  
  
        int getCountOfItems()  
        {  
            return countOfItems;  
        }  
  
        void setCountOfItems(int b)  
        {  
            this->countOfItems=b;  
        }  
  
        float getAmount()  
        {  
            return amount;  
        }  
  
        void setAmount(float c)  
        {  
            this->amount=c;  
        }  
  
        float getTotalAmt()
```

```

    {
        return totalAmt;
    }

    void setTotalAmt(float d)
    {
        this->totalAmt=d;
    }

    int getTotalCountOfItems()
    {
        return totalCountOfItems;
    }

    void settotalCountOfItems(int e)
    {
        this->totalCountOfItems=e;
    }

    void display(Purchase obj[],int n){
        int it_sum = 0;
        float pur_sum = 0;
        for(int i=0; i<n;i++)
        {
            pur_sum += obj[i].getAmount();
            it_sum += obj[i].getCountOfItems();
        }
        cout<<"\nPurchase Details :";
        for(int i=0; i<n; i++)
        {
            cout<<"\nCustomer "<<i+1<<" :"<<obj[i].getUserName();
            cout<<"\nNo of Items purchased :"<<obj[i].getCountOfItems();
            cout<<"\nPurchase amount :"<<obj[i].getAmount();
        }
        cout<<"\nTotal Amount Received :"<<pur_sum;
        cout<<"\nTotal Number of Items sold :"<<it_sum;
    }
};

int main()
{
    int num;
    cout<<"\nEnter the Number of customers :";
    cin>>num;
    Purchase p[10];
    string a;
    int b;
    float c;
    for(int i=0; i<num; i++)
    {

```

```

        cout<<"\nEnter the Name of the customer :";
        cin>>a;
        p[i].setUserName(a);
        cout<<"\nEnter the No of Items purchased :";
        cin>>b;
        p[i].setCountOfItems(b);
        cout<<"\nEnter the purchase amount :";
        cin>>c;
        p[i].setAmount(c);
    }

    p[0].display(p, num);
    return 0;
}

```

Constructors and Destructors

Problem 1: Consctructors

Main.cpp

```

#include <cstring>
#include<iostream>
#include<string>
#include "Item.cpp"
using namespace std;
int main()
{
    int choice;
    char ch='y';
    cout<<"\nMenu:";
    cout<<"\n1.Electronics";
    cout<<"\n2.Others";
    cout<<"\nEnter your choice:\n";
    cin>>choice;
    switch(choice)
    {
        case 1:
            {
                Item i1;
                string itemId;
                string itemName;
                cout<<"\nEnter details of the item";
                cout<<"\nEnter item id:"<<endl;
                cin>>itemId;
                i1.setItemId(itemId);
                cout<<"\nEnter the item name:"<<endl;
                cin>>itemName;
                i1.setItemName(itemName);
                cout<<"\nItem details";
                cout<<"\nItem id: "<<i1.getItemId();
                cout<<"\nItem name: "<<i1.getItemName();
                cout<<"\nItem type: "<<i1.getItemType();
                cout<<"\nItem vendor: "<<i1.getItemVendor();
            }
        }
    }
}

```

```

    }
    break;
case 2:
{
    Item i1;
    string temp;
    cout<<"\nEnter details of the item";
    cout<<"\nEnter item id:"<<endl;
    cin>>temp;
    i1.setItemId(temp);
    cout<<"\nEnter the item name:"<<endl;
    cin>>temp;
    i1.setItemName(temp);
    cout<<"\nEnter the item type:"<<endl;
    cin>>temp;
    i1.setItemType(temp);
    cout<<"\nEnter the item vendor:"<<endl;
    getline(cin, temp);
    getline(cin, temp);
    i1.setItemVendor(temp);
    cout<<"\nItem details";
    cout<<"\nItem id: "<<i1.getItemId();
    cout<<"\nItem name: "<<i1.getItemName();
    cout<<"\nItem type: "<<i1.getItemType();
    cout<<"\nItem vendor: "<<i1.getItemVendor();
    }
    break;
default:
    cout<<"\nInvalid choice";
    break;
}

return 0;
}

```

item.cpp

```

#include <cstring>
#include<iostream>
#include<string>
#include<stdio.h>

using namespace std;

class Item
{
    private:
        string itemId,itemName,itemType,itemVendor;
    public:
        string getItemId()
        {
            return itemId;
        }
        string getItemName()

```

```

    {
        return itemName;
    }
    string getItemType()
    {
        return itemType;
    }
    string getItemVendor()
    {
        return itemVendor;
    }
    void setItemId(string id)
    {
        itemId=id;
    }
    void setItemName(string name)
    {
        itemName=name;
    }
    void setItemType(string type)
    {
        itemType=type;
    }
    void setItemVendor(string vendor)
    {
        itemVendor=vendor;
    }
}

Item()
{
    itemType = "Electricals";
    itemVendor = "Arun electricals";
}
Item(string id,string name,string type,string vendor)
{
    itemId = id;
    itemName = name;
    itemType = type;
    itemVendor = vendor;
}
};

```

Problem 2: Destructors

Main.cpp

```

#include <cstring>
#include<iostream>
#include<string>
#include "Donor.cpp"
using namespace std;
int main()
{
    string name;
    float height, weight;

```



```

    int age, nof;
    cout<<"Welcome to the Blood Bank"<<endl<<"Enter the donor details"<<endl;
    cout<<"Enter the Name :"<<endl;
    cin>>name;
    cout<<"Enter the Age :"<<endl;
    cin>>age;
    cout<<"Enter the height :"<<endl;
    cin>>height;
    cout<<"Enter the weight :"<<endl;
    cin>>weight;
    cout<<"Enter the No of units donated :"<<endl;
    cin>>nof;
    Donor d(name, age, nof, height, weight);
    d.display();
}

```

Donor.cpp

```

#include <cstring>
#include<iostream>
#include<string>
#include<stdio.h>
#include <iomanip>

using namespace std;

class Donor
{
public:
    string name;
    int age,no_of_units_donated;
    float height,weight;
    Donor() {
        cout<<"Welcome to the Blood Bank"<<endl;
    }
    Donor(string n, int a, int no, float h, float w)
    {
        this->name = n;
        this->age = a;
        this->no_of_units_donated = no;
        this->height = double(h);
        this->weight = w;
    }
    void display() {
        cout<<"Donor details:\n";

        cout<<name<<endl<<age<<endl<<height<<endl<<fixed<<setprecision(1)<<weight<<endl<<no_of_units_donated<<endl;
    }
    ~Donor() {
        cout<<"Thank you for donating the Blood";
    }
};

```

Problem 3: Array of Objects

Main.cpp

```
#include <cstring>
#include<iostream>
#include<string>
#include "ItemTypeBO.cpp"
using namespace std;
int main()
{
    int num;
    cout<<"Enter the number of Itemtypes:";
    cin>>num;
    if(num < 1)
    {
        cout<<endl<<"Invalid Number";
        return 0;
    }
    ItemType I[10];
    string name;
    double deposit;
    double cpd;
    cout<<endl;
    for(int i = 0; i < num; i++)
    {
        ItemTypeBO itb;
        cout<<"Enter details of item type "<<i + 1;
        cout<<endl<<"Enter the Itemtype name:";
        getline(cin, name);
        getline(cin, name);
        cout<<endl<<"Enter the deposit:";
        cin>>deposit;
        cout<<endl<<"Enter cost per day:";
        cin>>cpd;
        cout<<endl;
        I[i] = itb.createItemType(name, deposit, cpd);
    }
    string search;
    cout<<"Enter the item name to be searched:";
    getline(cin, search);
    getline(cin, search);
    ItemTypeBO it;
    for(int i = 0; i < num; i++)
    {
        if(it.searchItemTypeByName(I, num, search) == 1)
        {
            cout<<endl<<"ItemType found";
            break;
        }
        else
        {
            cout<<endl<<"ItemType not found";
            break;
        }
    }
}
```

```
}  
}
```

ItemType.cpp

```
#include <cstring>  
#include<iostream>  
#include<string>  
  
using namespace std;  
  
class ItemType  
{  
    private:  
        string itemName;  
        double itemDeposit, costPerDay;  
    public :  
        string getItemName()  
        {  
            return itemName;  
        }  
        double getItemDeposit()  
        {  
            return itemDeposit;  
        }  
        double getCostPerDay()  
        {  
            return costPerDay;  
        }  
        void setItemName(string Name1)  
        {  
            this->itemName=Name1;  
        }  
        void setItemDeposit(double Deposit1)  
        {  
            this->itemDeposit=Deposit1;  
        }  
        void setCostPerDay(double costperday1)  
        {  
            this->costPerDay=costperday1;  
        }  
};
```

ItemTypeBo.cpp

```
#include <cstring>  
#include<iostream>  
#include<string>  
#include "ItemType.cpp"  
  
class ItemTypeBO  
{  
    public:  
        ItemType createItemType(string itemName, double itemDeposit, double  
costPerDay)  
        {  
            ItemType i;  
            i.setItemName(itemName);
```

```

        i.setItemDeposit(itemDeposit);
        i.setCostPerDay(costPerDay);
        return i;
    }

    int searchItemTypeByName(ItemType *itemtype, int n, string searchName)
    {
        for(int i = 0; i < n; i++)
        {
            if(itemtype[i].getItemName() == searchName)
                return 1;
        }
        return 0;
    }
};

```

Problem 4: FRIEND FUNCTION TO VALIDATE PASSWORD

Main.cpp

```

#include <iostream>
#include<string.h>
#include "User.cpp"
using namespace std;
string validate(string uname,string pword){
    User u1;
    User *uarr = u1.getUserDetails();
    for(int i = 0;i < 5; i++)
    {
        if(uname == uarr[i].getUserName())
        {
            if(pword == uarr[i].getPassword())
            {
                return "Yes";
            }

            return "No";
        }

        else
            continue;
    }
    return "No";
}

int main() {
    string name;
    string uname;
    string password;
    cout<<"Enter name:";
    cin>>name;
    cout<<endl<<"Enter the username:";
    cin>>uname;
}

```

```

        cout<<endl<<"Enter the password:";
        cin>>password;
        cout<<endl;
        string val = validate(uname, password);
        User u(name, uname, password);
        u.display(val);
    }

```

User.cpp

```

#include <iostream>
#include<string.h>
using namespace std;
class User{
    private:
        string name;
        string userName;
        string password;
    public:
        User()
        {

        }
        User(string n,string un,string p)
        {
            this->name = n;
            this->userName = un;
            this->password = p;
        }
        void setName(string n)
        {
            name=n;
        }
        void setUserName(string un)
        {
            userName=un;
        }
        void setPassword(string p)
        {
            password=p;
        }
        string getName()
        {
            return name;
        }
        string getUserName()
        {
            return userName;
        }
        string getPassword()
        {
            return password;
        }

        User * getUserDetails(){
            User *user = new User[5];

```

```

        user[0]= User("Abi","Abinaya","abi123");
        user[1]= User("Arun","Arunsoorya","arun456");
        user[2]= User("Sbi","Sbiharan","sbi789");
        user[3]= User("Sidhu","Siddarth","sid123");
        user[4]= User("Vivi","Viveka","vivi456");
        return user;
    }
    //Fill code

    void display(string s) {
        if(s=="Yes")
        {
            cout<<"Hiii..."<<this->name<<" !! Welcome to the event!!! ";
        }
        else
        {
            cout<<"Invalid username/password";
        }
    }

    friend string validate(string userName, string password);
};

```

Problem 5: Friend Class

Main.cpp

```

#include <iostream>
#include<string>
#include "EventExport.cpp"
using namespace std;
int main()
{
    int num, choice;
    cout<<"Enter the number of events :";
    cin>>num;
    string s;
    Event E[10];
    for(int i = 0; i< num; i++)
    {
        string name, detail, type, organiser;
        int attendeesCount;
        double projectedExpenses;
        cout<<endl<<"Enter the details of event "<<i+1<<endl;
        cout<<"Name :";
        getline(cin, name);
        getline(cin, name);
        cout<<endl<<"Detail :";
        getline(cin, detail);
        cout<<endl<<"Type :";
        cin>>type;
    }
}

```

```

        cout<<endl<<"Organiser :";
        cin>>organiser;
        cout<<endl<<"Attendees Count :";
        cin>>attendeesCount;
        cout<<endl<<"Projected Expenses :";
        cin>>projectedExpenses;
        Event k(name, detail, type, organiser, attendeesCount, projectedExpenses);
        E[i] = k;
    }
    cout<<endl<<"1. Export all the event details in CSV format";
    cout<<endl<<"2. Export name and organiser of the given event type";
    cout<<endl;
    cin>>choice;
    EventExport ee;
    switch(choice){
        case 1: ee.exportCSVFormat(E, num);
                break;
        case 2: cout<<"Enter the type";
                cin>>s;
                cout<<endl;
                ee.exportNameOrganiserInfo(E, num, s);
                break;
        default: cout<<"Invalid choice";
    }
    return 0;
}

```

Event.cpp

```

#include <iostream>
using namespace std;
class Event{
    private:
        string name;
        string details;
        string type;
        string organiser;
        int attendeesCount;
        double projectedExpenses;
    public:
        Event(){ }
        Event(string name, string detail, string type, string organiser, int
attendeesCount, double projectedExpenses){
            this->name = name;
            this->details = detail;
            this->type = type;
            this->organiser = organiser;
            this->attendeesCount = attendeesCount;
            this->projectedExpenses = projectedExpenses;
        }
        void setName(string name){
            this->name = name;
        }
        void setDetail(string detail){

```

```

        this->details = detail;
    }
    void setType(string type){
        this->type = type;
    }
    void setOrganiser(string organiser){
        this->organiser = organiser;
    }
    void setAttendeesCount(int attendeesCount){
        this->attendeesCount = attendeesCount;
    }
    void setProjectedExpenses(double projectedExpenses){
        this->projectedExpenses = projectedExpenses;
    }
    string getName(){
        return name;
    }
    string getDetail(){
        return details;
    }
    string getType(){
        return type;
    }
    string getOrganiser(){
        return organiser;
    }
    int getAttendeesCount(){
        return attendeesCount;
    }
    double getProjectedExpenses(){
        return projectedExpenses;
    }
    // fill the code
};

```

EventExport.cpp

```

#include <iostream>
#include <string>
#include "Event.cpp"
using namespace std;
class EventExport{
public:
    void exportCSVFormat(Event e[],int num)
    {
        for(int i = 0; i<num; i++)
        {
            cout<<e[i].getName()<<"", "<<e[i].getDetail()<<"", "<<e[i].getType()<<"",
"<<e[i].getOrganiser()<<"", "<<e[i].getAttendeesCount()<<"",
"<<e[i].getProjectedExpenses();
            cout<<endl;
        }
    }

    void exportNameOrganiserInfo(Event e[],int num,string type)
    {

```



```

        for(int i = 0; i<num; i++)
        {
            if(e[i].getType() == type)
            {
                cout<<e[i].getName()<<"", "<<e[i].getOrganiser()<<endl;
            }
        }
    }
};

```

File Handling

Problem 1: READING FROM FILE

Main.cpp

```

#include <iostream>
#include <fstream>
#include <string>
using namespace std;

int main() {
    string fname;
    string line;
    cout<<"Give filename:"<<endl;
    cin>>fname;
    ifstream ifile;
    ifile.open(fname);
    if(ifile.fail())
    {
        cout<<endl<<"Error! Incorrect file.";
    }
    else
    {
        while(getline(ifile, line))
        {
            cout<<line<<endl;
        }
    }
}

```

Problem 2: Writing data into a file in CSV format

Main.cpp

```

#include<iostream>
#include<string>
#include<stdio.h>
#include<fstream>
#include<list>
#include<iterator>
#include<sstream>
#include"UserBO.cpp"

```

```

using namespace std;
int main()
{
    int num;
    cout<<"Enter the number of users:";
    cin>>num;
    string name;
    string contact;
    string uname;
    string password;
    User U[10];
    for(int i = 0; i < num; i++)
    {
        cout<<endl<<"Enter the name of user :";
        cin>>name;
        cout<<endl<<"Enter the contact number :";
        cin>>contact;
        cout<<endl<<"Enter the username :";
        cin>>uname;
        cout<<endl<<"Enter the password :";
        cin>>password;
        User k(name, uname, password, contact);
        U[i] = k;
    }
    ofstream file;
    file.open("example.txt",ios::out);
    UserBO ub;
    ub.writeUserdetails(file, U, num);
    return 0;
}

```

User.cpp

```

#include<iostream>
#include<string>
#include<stdio.h>
#include<fstream>
#include<list>
#include<iterator>
#include<sstream>
using namespace std;
class User
{
private:
    string name;
    string username;
    string password;
    string contactnumber;
public:
    User(){}
    User(string name, string username, string password, string contactnumber)
    {
        this->name = name;
        this->username = username;
        this->password = password;
        this->contactnumber = contactnumber;
    }
}

```

```

    }
    void setName(string name)
    {
        this->name = name;
    }
    void setUsername(string uname)
    {
        this->username = uname;
    }
    void setPassword(string pass)
    {
        this->password = pass;
    }
    void setContactNumber(string connum)
    {
        this->contactnumber = connum;
    }
    string getName()
    {
        return name;
    }
    string getUsername()
    {
        return username;
    }
    string getPassword()
    {
        return password;
    }
    string getContactNumber()
    {
        return contactnumber;
    }
};

```

UserBo.cpp

```

#include<iostream>
#include<string>
#include<stdio.h>
#include<fstream>
#include<list>
#include<iterator>
#include<sstream>
#include"User.cpp"
using namespace std;
class UserBO
{
public:
    void writeUserdetails(ofstream &file, User obj[], int m)
    {
        for(int i = 0; i < m; i++)
        {

```

```

file<<obj[i].getName()<<","<<obj[i].getContactNumber()<<","<<obj[i].getUsername()<<
","<<obj[i].getPassword()<<endl;
    }
    file.close();
}
};

```

Problem 3: Reading and writing to a file

Main.cpp

```

#include <iostream>
#include <fstream>
#include<string.h>
#include<sstream>
#include<list>
using namespace std;
string* splitString(string str)
{
    string* arr = new string[25];
    int i=0;
    string delimiter = ",";
    size_t pos = 0;
    string token;
    str+=",";
    while ((pos = str.find(delimiter)) != string::npos)
    {
        token = str.substr(0, pos);
        arr[i]=token;
        str.erase(0, pos + delimiter.length());
        i++;
    }
    return arr;
}
int main()
{
    ifstream ifile;
    ofstream ofile;
    string line, eventName, type, eventDetail, eventOrganiser;
    ifile.open("readeventdetails.txt");
    ofile.open("writeeventdetails.txt");
    while(getline(ifile, line))
    {
        ofile<<line<<endl;
    }
    ifile.close();
    ofile.close();
    return 0;
}

```

EventType.cpp

```
#include <iostream>
#include <fstream>
#include<string.h>
#include<sstream>
#include<list>
using namespace std;
class EventType
{
    private:
        string type;
    public:
        EventType() {
        }
        EventType(string type){
            this->type = type;
        }
        string getType() {
            return type;
        }
};
```

Event.cpp

```
#include <iostream>
#include <fstream>
#include<string.h>
#include<sstream>
#include<list>
#include "EventType.cpp"
using namespace std;
class Event
{
    private:
        string eventName;
        EventType type;
        string eventDetail;
        string eventOrganiser;
    public:
        Event() {}
        Event(string eventName, EventType type, string evedet, string eveorg)
        {
            this->eventName = eventName;
            this->type = type;
            this->eventDetail = evedet;
            this->eventOrganiser = eveorg;
        }
};
```

Problem 4: File Details

Main.cpp

```
#include<iostream>
#include<string.h>
```

```

#include<fstream>
using namespace std;
int main ()
{
    string line;
    string filename;
    cout<<"Enter the file name :";
    cin>>filename;
    ifstream ifile;
    ifile.open(filename);
    int count = 0, i;
    int lin = 0;
    while(getline(ifile, line))
    {
        for (i = 0; line[i] != '\0';i++)
        {
            if (line[i] == ' ')
                count++;
        }
        lin++;
    }

    cout<<endl<<"Number of words in file : "<<count + lin;
    cout<<endl<<"Numbers of lines in the file : "<< lin;
    return 0;
}

```

Inheritance

Simple Problems

Problem 1: SINGLE INHERITANCE

Main.cpp

```

#include <iostream>
using namespace std;
#include "Developer.cpp"
int main(void)
{
    string name,skills,role;
    int empId,experience;
    cout<<"Enter the name";
    cin>>name;
    cout<<endl<<"Enter the experience";
    cin>>experience;
    cout<<endl<<"Enter the employee id";
    cin>>empId;
    cout<<endl<<"Enter the technical skills of the developer";
    cin>>skills;
    cout<<endl<<"Enter the role of the developer";
    cin>>role;
    Developer D(name, experience, empId, skills, role);
    D.displayDetails();
}

```

```
    return 0;
}
```

Employee.h

```
#include<iostream>
using namespace std;
class Employee{
    protected:
        string name;
        int experience;
        int empId;
    public:
        Employee() {}

        Employee(string name,int experience, int empId)
        {
            this->name=name;
            this->experience=experience;
            this->empId=empId;
        }

};
```

Developer.cpp

```
#include<iostream>
using namespace std;
#include "Employee.h"
class Developer: public Employee{
    private:
        string skills;
        string role;

    public:
        Developer() {}

        Developer(string name,int experience,int empId,string skills,string
role):Employee(name,experience,empId)
        {
            this->skills=skills;
            this->role=role;
        }

        void displayDetails()
        {
            cout<<endl<<"Employee Details";
            cout<<endl<<"Name : "<<this->name;
            cout<<endl<<"Experience : "<<this->experience<<" years";
            cout<<endl<<"Employee Id : "<<this->empId;
            cout<<endl<<"Technical Skills : "<<this->skills;
            cout<<endl<<"Role : "<<this->role;
        }

};
```

Problem 2: HIERARCHICAL INHERITANCE

Main.cpp

```
#include <iostream>
using namespace std;
#include "EBook.cpp"

int main(void)
{
    string diskType,name,author,publication,paperType;
    int size,price,numOfPages;
    int ch;
    cout<<"Enter the type of book";
    cout<<endl<<"1.PrintedBook";
    cout<<endl<<"2.EBook";
    cin>>ch;
    switch(ch)
    {
        case 1:{ cout<<endl<<"Enter the book name";
            getline(cin, name);
            getline(cin, name);
            cout<<endl<<"Enter the author name";
            getline(cin, author);
            cout<<endl<<"Enter the price:";
            cin>>price;
            cout<<endl<<"Enter the publication name:";
            cin>>publication;
            cout<<endl<<"Enter the number of pages of the book";
            cin>>numOfPages;
            cout<<endl<<"Enter the paper type of the book";
            cin>>paperType;
            PrintedBook P(name, author, price, publication, numOfPages,
paperType);
            P.displayDetails();
            break; }
        case 2: { cout<<endl<<"Enter the book name";
            getline(cin, name);
            getline(cin, name);
            cout<<endl<<"Enter the author name";
            getline(cin, author);
            cout<<endl<<"Enter the price:";
            cin>>price;
            cout<<endl<<"Enter the publication name:";
            cin>>publication;
            cout<<endl<<"Enter the disk type of the book";
            cin>>diskType;
            cout<<endl<<"Enter the size of the disk";
            cin>>size;
            EBook E(name, author, price, publication, diskType, size);
            E.displayDetails();
            break; }
    }
    return 0;
}
```


Book.h

```
class Book{
    protected:
        string name;
        string author;
        int price;
        string publication;

    public:
        Book() {}

        Book(string name, string author, int price, string publication)
        {
            this->name=name;
            this->author=author;
            this->price=price;
            this->publication=publication;
        }

};
```

PrintedBook.cpp

```
#include<iostream>
using namespace std;
#include "Book.h"
class PrintedBook: public Book{
    private:
        int numOfPages;
        string paperType;

    public:
        PrintedBook() {}
        PrintedBook(string name, string author, int price, string publication, int
numOfPages, string paperType):Book(name, author, price, publication)
        {
            this->numOfPages = numOfPages;
            this->paperType = paperType;
        }

        void displayDetails()
        {
            cout<<endl<<"Printed Book Details";
            cout<<endl<<"Name : "<<this->name;
            cout<<endl<<"Author : "<<this->author;
            cout<<endl<<"Price : "<<this->price;
            cout<<endl<<"Publication : "<<this->publication;
            cout<<endl<<"Number Of Pages : "<<this->numOfPages;
            cout<<endl<<"Paper Type : "<<this->paperType;
        }

};
```

EBook.cpp

```
#include<iostream>
```

```

using namespace std;
#include "PrintedBook.cpp"
class EBook: public Book{
    private:
        string diskType;
        int size;

    public:
        EBook(){}
        EBook(string name, string author, int price, string publication, string
diskType, int size):Book(name,author,price,publication)
        {
            this->diskType=diskType;
            this->size=size;
        }

        void displayDetails()
        {
            cout<<endl<<"EBook Details";
            cout<<endl<<"Name : "<<this->name;
            cout<<endl<<"Author : "<<this->author;
            cout<<endl<<"Price : "<<this->price;
            cout<<endl<<"Publication : "<<this->publication;
            cout<<endl<<"Disk Type : "<<this->diskType;
            cout<<endl<<"Size : "<<this->size<<" MB";
        }
};

```

Problem 3: MULTILEVEL INHERITANCE

Main.cpp

```

#include <iostream>
using namespace std;
#include "Developer.cpp"
int main()
{
    string name,skills,role,gender;
    int empId,experience;
    cout<<"Enter the name";
    cin>>name;
    cout<<endl<<"Enter the gender";
    cin>>gender;
    cout<<endl<<"Enter the experience";
    cin>>experience;
    cout<<endl<<"Enter the employee id";
    cin>>empId;
    cout<<endl<<"Enter the technical skills of the developer";
    cin>>skills;
    cout<<endl<<"Enter the role of the developer";
    cin>>role;
    Developer D(name, gender, experience, empId, skills, role);
    D.displayDetails();
    return 0;
}

```

```
}
```

Person.h

```
#include<iostream>
using namespace std;
class Person{
    protected:
        string name;
        string gender;
    public:
        Person(){}

        Person(string name,string gender)
        {
            this->name=name;
            this->gender=gender;
        }
};
```

Developer.cpp

```
#include<iostream>
using namespace std;
#include "Employee.h"
class Developer: public Employee{
    private:
        string skills;
        string role;

    public:
        Developer(){}

        Developer(string name,string gender,int experience,int empId,string
skills,string role):Employee(name,gender,experience,empId)
        {
            this->skills = skills;
            this->role = role;
        }

        void displayDetails()
        {
            cout<<endl<<"Developer Details";
            cout<<endl<<"Name : "<<<this->name;
            cout<<endl<<"Gender : "<<<this->gender;
            cout<<endl<<"Experience : "<<<this->experience<<" years";
            cout<<endl<<"Employee Id : "<<<this->empId;
            cout<<endl<<"Technical Skills : "<<<this->skills;
            cout<<endl<<"Role : "<<<this->role;
        }
};
```

Employee.h

```
#include<iostream>
using namespace std;
#include "Person.h"
class Employee:public Person{
    protected:
        int experience;
        int empId;
    public:
        Employee(){}

        Employee(string name,string gender,int experience, int
empId):Person(name,gender)
        {
            this->experience=experience;
            this->empId=empId;
        }

};
```

Problem 4: MULTIPLE INHERITANCE

Main.cpp

```
#include <iostream>
using namespace std;
#include "Truck.cpp"

int main(void)
{
    string model,manufacturer,gearType,fuelType,size;
    int year,cargoCapacity;
    cout<<"Enter the model of the vehicle";
    cin>>model;
    cout<<endl<<"Enter the manufactured year";
    cin>>year;
    cout<<endl<<"Enter the name of the manufacturer";
    getline(cin, manufacturer);
    getline(cin, manufacturer);
    cout<<endl<<"Enter the gear type of the four wheeler";
    cin>>gearType;
    cout<<endl<<"Enter the fuel type of the four wheeler";
    cin>>fuelType;
    cout<<endl<<"Enter the cargo capacity of the truck";
    cin>>cargoCapacity;
    cout<<endl<<"Enter the size of the truck";
    cin>>size;
    Truck T(model, year, manufacturer, gearType, fuelType, cargoCapacity, size);
    T.displayDetails();
    return 0;
}
```

FourWheel.h

```
#include<iostream>
using namespace std;
class FourWheeler{
    protected:
        string gearType;
        string fuelType;

    public:
        FourWheeler(){}
        FourWheeler(string gearType, string fuelType)
        {
            this->gearType=gearType;
            this->fuelType=fuelType;
        }
};
```

Truck.cpp

```
#include<iostream>
using namespace std;
#include "FourWheeler.h"
#include "Vehicle.h"

class Truck: public FourWheeler, public Vehicle{
    private:
        int cargoCapacity;
        string size;

    public:
        Truck(){}
        Truck(string model, int year, string manufacturer, string gearType, string fuelType, int cargoCapacity, string size):FourWheeler(gearType, fuelType),Vehicle(model, year, manufacturer)
        {
            this->cargoCapacity = cargoCapacity;
            this->size = size;
        }

        void displayDetails()
        {
            cout<<endl<<"Truck Details";
            cout<<endl<<"Model : "<<this->model;
            cout<<endl<<"Year : "<<this->year;
            cout<<endl<<"Manufacturer : "<<this->manufacturer;
            cout<<endl<<"Gear Type : "<<this->gearType;
            cout<<endl<<"Fuel Type : "<<this->fuelType;
            cout<<endl<<"Cargo Capacity : "<<this->cargoCapacity<<" MT";
            cout<<endl<<"Size : "<<this->size;
        }
};
```

Vehicle.h

```
#include<iostream>
using namespace std;
class Vehicle{
    protected:
        string model;
        int year;
        string manufacturer;

    public:
        Vehicle(){}
        Vehicle(string model, int year, string manufacturer)
        {
            this->model=model;
            this->year=year;
            this->manufacturer=manufacturer;
        }
};
```

Additional Problems

Problem 1: INHERITANCE

Main.cpp

```
#include <iostream>
#include "SavingsAccount.cpp"
#include "CurrentAccount.cpp"
#include "Account.h"
#include <sstream>
using namespace std;
string* splitString(string str){
    string* arr = new string[25];
    int i=0;
    string delimiter = ",";
    size_t pos = 0;
    string token;
    str+=",";
    while ((pos = str.find(delimiter)) != string::npos) {
        token = str.substr(0, pos);
        arr[i]=token;

        str.erase(0, pos + delimiter.length());
        i++;
    }

    return arr;
}
int main(){
    int choice;
```

```

//fill code
cout<<"Choose the account type"<<endl;
cout<<"1.Savings Account"<<endl;
cout<<"2.Current Account"<<endl;
cout<<"Enter your choice"<<endl;
cin>>choice;
cin.get();
if(choice==1)
{
    string values;
    cout<<"Enter Account details in comma separated(Account Holder Name,Account
Number,Bank Name,Organisation Name)"<<endl;
    getline(cin, values);
    string *arr = splitString(values);
    SavingsAccount sa;
    sa.setAccName(arr[0]);
    sa.setAccNo(arr[1]);
    sa.setBankName(arr[2]);
    sa.setOrgName(arr[3]);
    sa.display();
}
else
{
    cout<<"Enter Account details in comma separated(Account Holder Name,Account
Number,Bank Name,TIN Number)"<<endl;
    string values;
    getline(cin, values);
    string *arr = splitString(values);
    CurrentAccount ca;
    ca.setAccName(arr[0]);
    ca.setAccNo(arr[1]);
    ca.setBankName(arr[2]);
    ca.setTinNumber(arr[3]);
    ca.display();
}
}

```

Accounts.h

```

#ifndef MYHEADER_H
#define MYHEADER_H
#include <iostream>
using namespace std;
class Account{
protected:
    string accNo;
    string accName;
    string bankName;
public:
    void setAccNo (string accNo){
        this->accNo = accNo;
    }
    void setAccName (string accName){

```

```

        this->accName = accName;
    }
    void setBankName (string bankName){
        this->bankName = bankName;
    }
    string getAccNo(){
        return accNo;
    }
    string getAccName(){
        return accName;
    }
    string getBankName(){
        return bankName;
    }
};
#endif

```

SavingsAccount.cpp

```

#include <iostream>
#include "Account.h"
using namespace std;
class SavingsAccount : public Account
{
    private:
        string orgName;
    public:
        void setOrgName(string orgName){
            this->orgName = orgName;
        }
        string getOrgName(){
            return orgName;
        }
        void display(){
            cout<<"Account holder name:"<<getAccName();
            cout<<endl<<"Account number:"<<getAccNo();
            cout<<endl<<"Bank name:"<<getBankName();
            cout<<endl<<"Organisation name:"<<getOrgName();
        }
};

```

CurrentAccount.cpp

```

#include <iostream>
#include "Account.h"
using namespace std;
class CurrentAccount : public Account
{
    private:
        string tinNumber;
    public:

```



```

        void setTinNumber(string tinNumber){
            this->tinNumber = tinNumber;
        }
        string getTinNumber(){
            return tinNumber;
        }
        void display(){
            cout<<"Account holder name:"<<getAccName();
            cout<<endl<<"Account number:"<<getAccNo();
            cout<<endl<<"Bank name:"<<getBankName();
            cout<<endl<<"TIN number:"<<getTinNumber();
        }
    };

```

Problem 2: SUPER() KEYWORD

Main.cpp

```

#include <iostream>
#include<stdio.h>
#include<string.h>
#include<sstream>
#include<stdlib.h>
#include<cstdlib>
#include<iomanip>
#include"Exhibition.cpp"
#include"StageEvent.cpp"
using namespace std;
string* splitString(string str){
    string* arr = new string[25];
    int i=0;
    string delimiter = ",";
    size_t pos = 0;
    string token;
    str+=",";
    while ((pos = str.find(delimiter)) != string::npos) {
        token = str.substr(0, pos);
        arr[i]=token;

        str.erase(0, pos + delimiter.length());
        i++;
    }
    return arr;
}
int main()
{
    int choice;
    int i=0;
    string str,*str1;
    string exhibitionDetail, stageEvent;
    int days;
    int x,y;
    double totalcost,cost;
    cout<<"Event List\n";

```

```

cout<<"Press 1 for Exhibition\n";
cout<<"Press 2 for Stage Event\n";
cin>>choice;
switch(choice)
{
    case 1:
        {
            string values;
            cout<<"Enter the details of Exhibition";
            getline(cin, values);
            getline(cin, values);
            string *arr = splitString(values);
            stringstream seats(arr[4]);
            stringstream cost(arr[5]);
            int seat;
            double costpd;
            seats >> seat;
            cost >> costpd;
            Exhibition ex(arr[0], arr[1], arr[2], arr[3], seat, costpd);
            int days;
            cout<<endl<<"Enter the total number of days:";
            cin>>days;
            cout<<endl;
            cout<<"Exhibition"<<endl;
            ex.display(days);
            break;
        }
    case 2:
        {
            string values;
            cout<<"Enter the details of Stage Event";
            getline(cin, values);
            getline(cin, values);
            string *arr = splitString(values);
            stringstream seats(arr[4]);
            stringstream cost(arr[5]);
            int seat;
            double costpd;
            seats >> seat;
            cost >> costpd;
            StageEvent sa(arr[0], arr[1], arr[2], arr[3], seat, costpd);
            int days;
            cout<<endl<<"Enter the total number of days:";
            cin>>days;
            cout<<endl;
            cout<<"Stage Event"<<endl;
            sa.display(days);
            break;
        }
}
return 0;
}

```

StageEvent.cpp

```
#include <iostream>
#include<stdio.h>
#include<iomanip>
#include"Event.h"
using namespace std;
class StageEvent:public Event
{
    private:
        int noOftickets;
    public:
        StageEvent(){}
        StageEvent(string name, string detail, string type, string organiser , int
noOftickets, double costPerDay):Event( name, detail, type, organiser, costPerDay)
        {
            this->noOftickets = noOftickets;
        }
        void setNoOftickets(int noOfSeats){
            this->noOftickets = noOfSeats;
        }
        int getNoOftickets(){
            return noOftickets;
        }
        double calculateCost(int days)
        {
            double norm = (getCostPerDay() * days)*1.15;
            return norm;
        }
        void display(int days){
            cout<<"Name : "<<getName();
            cout<<endl<<"Detail : "<<getDetail();
            cout<<endl<<"Type : "<<getType();
            cout<<endl<<"Organiser : "<<getOrganiser();
            cout<<endl<<"Number of Seats : "<<getNoOftickets();
            cout<<endl<<"Total cost for "<< days <<" days is :
Rs."<<fixed<<setprecision(2)<<calculateCost(days);
        }
};
```

Event.h

```
#ifndef MYHEADER_H
#define MYHEADER_H
#include <iostream>
#include<stdio.h>
using namespace std;
class Event{
    protected:
        string name;
        string detail;
        string type;
```

```

        string organiser;
        double costPerDay;
    public:
        Event() {}
        Event(string name, string detail, string type, string organiser, double
price)
        {
            this->name = name;
            this->detail = detail;
            this->type = type;
            this->organiser = organiser;
            this->costPerDay=price;
        }
        void setName(string name){
            this->name = name;
        }
        void setDetail(string detail){
            this->detail = detail;
        }
        void setType(string type){
            this->type = type;
        }
        void setOrganiser(string organiser){
            this->organiser = organiser;
        }

        double getCostPerDay()
        {
            return costPerDay;
        }
        void setCostPerDay(double price)
        {
            this->costPerDay=price;
        }
        string getName(){
            return name;
        }
        string getDetail(){
            return detail;
        }
        string getType(){
            return type;
        }
        string getOrganiser(){
            return organiser;
        }
    };
#endif

```

Exhibition.cpp

```

#include <iostream>
#include<stdio.h>
#include <iomanip>

```

```

#include "Event.h"
using namespace std;
class Exhibition: public Event
{
private:
    int noOfstalls;
public:
    Exhibition(){}
    Exhibition(string name, string detail, string type, string organiser , int
noOfstalls, double costPerDay):Event( name, detail, type, organiser, costPerDay)
    {
        this->noOfstalls = noOfstalls;
    }
    void setNoOfstalls(int noOfStalls){
        this->noOfstalls = noOfStalls;
    }
    int getNoOfstalls(){
        return noOfstalls;
    }
    double calculateCost(int totaldays)
    {
        double norm = getCostPerDay() * totaldays;
        return norm * 1.05;
    }
    void display(int days){
        cout<<"Name : "<<getName();
        cout<<endl<<"Detail : "<<getDetail();
        cout<<endl<<"Type : "<<getType();
        cout<<endl<<"Organiser : "<<getOrganiser();
        cout<<endl<<"Number of Stalls : "<<getNoOfstalls();
        cout<<endl<<"Total cost for "<< days <<" days is :
Rs."<<fixed<<setprecision(2)<<calculateCost(days);
    }
};

```

Problem 3: MULTILEVEL INHERITANCE

Main.cpp

```

#include<iostream>
#include<string>
#include "AccountBO.h"
using namespace std;
int main()
{
    AccountBO ab;
    FixedAccount fa;
    string details;
    cout<<"Enter account details"<<endl;
    getline(cin,details);
    fa = ab.accountDetails(details);
    fa.display();
}

```

FixedAccount.h

```
#ifndef HEADER
#define HEADER
#include<iostream>
#include<string>
#include<stdio.h>
#include<iomanip>
#include <bits/stdc++.h>
#include "SavingAccount.cpp"
using namespace std;
class FixedAccount: public SavingAccount
{
    private:
        int lockingPeriod;
    public:
        FixedAccount() {}
        FixedAccount(int lockingPeriod)
        {
            this->lockingPeriod=lockingPeriod;
        }
        void setLockingPeriod(int lockingPeriod)
        {
            this->lockingPeriod=lockingPeriod;
        }
        int getLockingPeriod()
        {
            return lockingPeriod;
        }
        void display()
        {
            cout<<"Account Details:"<<endl;
            printf("%-20s %-20s %-20s %-20s
%s\n", "AccountHolderName", "AccountNumber", "Balance", "MinimumBalance", "LockingPeriod
");
            string acchname = getAccountHolderName();
            int n = acchname.length();
            char acc_name[n + 1];
            strcpy(acc_name, acchname.c_str());
            string accno = getAccountNumber();
            n = accno.length();
            char acc_no[n + 1];
            strcpy(acc_no, accno.c_str());
            printf("%-20s %-20s %-20.2f %-20.2f
%d\n", acc_name, acc_no, getBalance(), getMinimumBalance(), getLockingPeriod());
        }
};
#endif
```

Account.cpp

```
#include<iostream>
#include<string>
using namespace std;
```

```

class Account
{
    protected:
        string accountHolderName;
        string accountNumber;
        double balance;
    public:
        Account() {}
        Account(string accountHolderName, string accountNumber, double balance)
        {
            this->accountHolderName=accountHolderName;
            this->accountNumber=accountNumber;
            this->balance=balance;
        }
        void setAccountHolderName(string accountHolderName)
        {
            this->accountHolderName=accountHolderName;
        }
        string getAccountHolderName()
        {
            return accountHolderName;
        }
        void setAccountNumber(string accountNumber)
        {
            this->accountNumber=accountNumber;
        }
        string getAccountNumber()
        {
            return accountNumber;
        }
        void setBalance(double balance)
        {
            this->balance=balance;
        }
        double getBalance()
        {
            return balance;
        }
};
//#endif

```

AccountBo.h

```

#include<iostream>
#include<string>
#include <stdlib.h>
#include<sstream>
#include "FixedAccount.h"
using namespace std;
string* splitString(string str){
    string* arr = new string[25];
    int i=0;
    string delimiter = ",";
    size_t pos = 0;

```

```

string token;
str+=",";
while ((pos = str.find(delimiter)) != string::npos) {
    token = str.substr(0, pos);
    arr[i]=token;

    str.erase(0, pos + delimiter.length());
    i++;
}

return arr;
}

class AccountBO{
public:
    FixedAccount fa;
    FixedAccount accountDetails(string details){
        string* det = splitString(details);
        stringstream balance(det[2]);
        stringstream minimbala(det[3]);
        stringstream lockperiod(det[4]);
        double bal, minbal;
        int lockped;
        balance >> bal;
        minimbala >> minbal;
        lockperiod >> lockped;
        fa.setAccountHolderName(det[0]);
        fa.setAccountNumber(det[1]);
        fa.setBalance(bal);
        fa.setMinimumBalance(minbal);
        fa.setLockingPeriod(lockped);
        return fa;
    }
};

```

SavingAccount.cpp

```

#include<iostream>
#include "Account.cpp"
using namespace std;
class SavingAccount: public Account
{
protected:
    double minimumBalance;
public:
    SavingAccount(){}
    SavingAccount(double minimumBalance)
    {
        this->minimumBalance=minimumBalance;
    }
    void setMinimumBalance(double minimumBalance)
    {
        this->minimumBalance=minimumBalance;
    }
}

```



```

    double getMinimumBalance()
    {
        return minimumBalance;
    }
};

```

Problem 4: SIMPLE MULTIPLE INHERITANCE

//other files don't need edit to match output

Main.cpp

```

#include<iostream>
#include<iomanip>
using namespace std;
#include "PetrolBO.cpp"
int main()
{
    float density, rate;
    int thermalConductivity;
    cout<<"Enter the details\n";
    cout<<"Enter the density of the petrol";
    cin>>density;
    cout<<endl<<"Enter the rate of the petrol";
    cin>>rate;
    cout<<endl<<"Enter the thermal conductivity of the petrol"<<endl;
    cin>>thermalConductivity;
    double gravity = density/997.0;
    cout<<"Density : "<<density;
    cout<<endl<<"Rate : "<<rate;
    cout<<endl<<"Thermal Conductivity : "<<thermalConductivity;
    cout<<endl<<"Gravity : "<<gravity;
    return 0;
}

```

Problem 5: MULTIPLE INHERITANCE

Main.cpp

```

#include <iostream>
#include<stdio.h>
#include<bits/stdc++.h>
#include"Circle.cpp"
#include"Rectangle.cpp"
#include"Square.cpp"
using namespace std;
int main()
{
    int input;
    double len,wid,rad;
    double a;
    Rectangle r;
    Circle c;
}

```

```

    Square s;
    cout<<"Select the shape:\n1.Circle\n2.Rectangle\n3.Square"<<endl;
    cin>>input;
    switch(input)
    {
        case 1: cout<<"Enter the radius of the circle:";
                cin>>rad;
                c.setRadius(rad);
                cout<<endl<<"Perimeter of Circle is:
"<<fixed<<setprecision(2)<<c.calculatePerimeter();
                break;
        case 2: cout<<"Enter the length of the rectangle:";
                cin>>len;
                cout<<endl<<"Enter the width of the rectangle:";
                cin>>wid;
                r.setLength(len);
                r.setWidth(wid);
                cout<<endl<<"Perimeter of Rectangle is:
"<<fixed<<setprecision(2)<<r.calculatePerimeter();
                break;
        case 3: cout<<"Enter the length of the square:";
                cin>>a;
                s.setLength(a);
                cout<<endl<<"Perimeter of Square is:
"<<fixed<<setprecision(2)<<s.calculatePerimeter();
                break;
        default:
                cout<<"Nothing";
    }

    return 0;
}

```

Circle.cpp

```

#include <iostream>
#include<stdio.h>
#include"Shape.h"
using namespace std;
class Circle: public Shape
{
    double radius;
public:
    double getRadius()
    {
        return radius;
    }
    void setRadius(double radius)
    {
        this->radius=radius;
    }
    double calculatePerimeter()

```

```

    {
        return 2*3.14*radius;
    }

};

```

Rectangle.cpp

```

#include <iostream>
#include<stdio.h>
#include"Shape.h"
using namespace std;
class Rectangle: public Shape
{
    double length;
    double breadth;
public:
    double getLength()
    {
        return length;
    }
    double getBreadth()
    {
        return breadth;
    }
    void setLength(double length)
    {
        this->length=length;
    }
    void setWidth(double breadth)
    {
        this->breadth=breadth;
    }
    double calculatePerimeter()
    {
        return 2*(length + breadth);
    }
};

```

Shape.h

```

#ifndef MYHEADER_H
#define MYHEADER_H
#include <iostream>
#include<stdio.h>
using namespace std;
class Shape
{
public:
    virtual double calculatePerimeter() = 0;
};

```

```
#endif
```

Square.cpp

```
#include <iostream>
#include<stdio.h>
#include"Shape.h"
using namespace std;
class Square: public Shape
{
    double length;
public:

    double getLength()
    {
        return length;
    }
    void setLength(double length)
    {
        this->length=length;
    }
    double calculatePerimeter()
    {
        return 4*length;
    }
};
```