**Question Number 1:**

**PROGRAM:**

#include <iostream>

using namespace std;

class Example

{

public:

Example(int y = 10) :data(y)

{

}

int getIncrementedData() // removing const from here

{ // if function is constant than we cannot increment in it.

return ++data;

}

static void getCount()

{

cout << "Count is " << count << endl;

}

// without any return type

// initiallising it outside the class

// static function can only access static variable but here data is not static

// putting count here rather than data;

private:

int data;

static int count;

};

int Example::count = 5;

int main()

{

Example E(13);

E.getIncrementedData();

cout << endl;

E.getCount();

cout << endl << endl;

system("pause");

}

**A screenshot of a computer screen

Description automatically generated**

**Question Number 2:**

**PROGRAM:**

#include <iostream>

using namespace std;

class Savings\_Account

{

public:

Savings\_Account(int a = 0)

{

savings\_Balance = a;

}

static void modify\_Interest\_Rate(float a)

{

annual\_Interest\_Rate = a; //setting interest values;

}

void calculate\_Monthly\_Interest()

{ // calculatin rates and adding them

savings\_Balance = savings\_Balance + ((savings\_Balance\*annual\_Interest\_Rate) / 12);

}

void setdata()

{

cout << savings\_Balance; //cout calculated data

}

private:

static float annual\_Interest\_Rate;

float savings\_Balance;

};

float Savings\_Account::annual\_Interest\_Rate = 0;

int main()

{

Savings\_Account saver1(2000), saver2(3000);

cout << "Initial Values :\nSaver 1 : $";

saver1.setdata();

cout << "\t\t Saver 2 : $";

saver2.setdata();

// at 0.03 interest

saver1.modify\_Interest\_Rate(.03);

saver2.modify\_Interest\_Rate(.03);

saver1.calculate\_Monthly\_Interest();

saver2.calculate\_Monthly\_Interest();

cout << endl <<endl << "Balance after 1 month's interest applied at .03 :\nSaver 1 : $";

saver1.setdata();

cout << "\t\t Saver 2 : ";

saver2.setdata();

// at 0.04 interest

saver1.modify\_Interest\_Rate(.04);

saver2.modify\_Interest\_Rate(.04);

saver1.calculate\_Monthly\_Interest();

saver2.calculate\_Monthly\_Interest();

cout << endl << endl << "Balance after 2 month's interest applied at .04 :\nSaver 1 : $";

saver1.setdata();

cout << "\t Saver 2 : $";

saver2.setdata();

cout << endl << endl;

system("pause");

}

**A screenshot of a computer screen

Description automatically generated**

**Question Number 3:**

**PROGRAM:**

#include <iostream>

#include <stdlib.h>

#include <string>

using namespace std;

class Package

{

public:

Package()

{

name = { 0 }, address = { 0 }, city = { 0 }, state = { 0 }, Zip\_Code = 0, cost = 1.5, weight = 0, Total = 0;

}

void getdata()

{

cout << "Enter The Name : ";

cin>>name;

cout << "Enter The Address : ";

cin>>address;

cout << "Enter The City : ";

cin>>city;

cout << "Enter The State : ";

cin>>state;

cout << "Enter The Zip\_Code : ";

cin >> Zip\_Code;

}

void setdata()

{

cout << endl << name;

cout << endl << address;

cout << endl << city;

cout << endl << state;

cout << endl << Zip\_Code << endl;

}

double calculate\_Cost()

{

cout << "Enter the Weight of Package : ";

cin >> weight;

Total = cost\*weight;

cout << "Cost per Ounce is $1.5\nAccording to this your total cost becomes : "<<Total;

return Total;

}

string name;

string address;

string city;

string state;

int Zip\_Code;

float cost;

float weight;

double Total;

};

class Two\_Day\_Package :public Package

{

public:

Two\_Day\_Package()

{

name = { 0 }, address = { 0 }, city = { 0 }, state = { 0 }, Zip\_Code = 0, cost = 1.5, weight = 0,Total = 0, flat\_fee=0.7;

}

double calculate\_Cost()

{

cout << "Enter the Weight of Package : ";

cin >> weight;

Total = (cost\*weight) + flat\_fee;

cout << "Cost per Ounce is $1.5\nHere you Chosed Two Day Package\nHence Total cost becomes : "<<Total;

return Total;

}

float flat\_fee;

};

class Overnight\_Package :public Package

{

public:

Overnight\_Package()

{

name = { 0 }, address = { 0 }, city = { 0 }, state = { 0 }, Zip\_Code = 0, cost = 1.5, weight = 0,Total = 0,additional\_Charges=1.5;

}

double calculate\_Cost()

{

cout << "Enter the Weight of Package : ";

cin >> weight;

Total = (cost\*weight) + additional\_Charges;

cout << "Cost per Ounce is $1.5\nHere you Chosed Over-Night Package\nHence Total cost will be according to $3 : "<<Total;

return Total;

}

float additional\_Charges;

};

int main()

{

Package Package[2], Package1[2];

Two\_Day\_Package Package2[2];

Overnight\_Package Package3[2];

cout << "Enter Data For Sender 1" << endl << endl;

Package1[0].getdata();

Package1[0].calculate\_Cost();

cout << endl << endl <<"Enter Data For Reciver 1" << endl << endl;

Package1[1].getdata();

cout << endl;

cout << "Enter Data For Sender 2" << endl << endl;

Package[0].getdata();

Package[0].calculate\_Cost();

cout << endl << endl << "Enter Data For Reciver 2" << endl << endl;

Package[1].getdata();

cout << endl;

cout << "Enter Data For Sender 3" << endl << endl;

Package2[0].getdata();

Package2[0].calculate\_Cost();

cout << endl << endl<< "Enter Data For Reciver 3" << endl << endl;

Package2[1].getdata();

cout << endl;

cout << "Enter Data For Sender 4" << endl << endl;

Package3[0].getdata();

Package3[0].calculate\_Cost();

cout << endl << endl << "Enter Data For Reciver 4" << endl << endl;

Package3[1].getdata();

cout << endl;

system("cls");

cout << " Sender 1 :" << endl << endl;

Package[0].setdata();

cout << endl << endl << "Reciver 1 :" << endl << endl;

Package[1].setdata();

cout << "Cost : $"<<Package[0].Total;

cout << endl;

cout << endl << " Sender 2 :" << endl << endl;

Package1[0].setdata();

cout << endl << endl << "Reciver 2 :" << endl << endl;

Package1[1].setdata();

cout << "Cost : $" << Package1[0].Total;

cout << endl;

cout << endl <<" Sender 3 :" << endl << endl;

Package2[0].setdata();

cout << endl << endl << "Reciver 3 :" << endl << endl;

Package2[1].setdata();

cout << "Cost : $" << Package2[0].Total;

cout << endl;

cout << endl <<" Sender 4 :" << endl << endl ;

Package3[0].setdata();

cout << endl << endl << "Reciver 4 :" << endl << endl ;

Package3[1].setdata();

cout << "Cost : $" << Package3[0].Total;

cout << endl << endl;

system("pause");

}

**A screenshot of a computer screen

Description automatically generated**

**A screenshot of a computer screen

Description automatically generated**

**Question Number 4:**

**PROGRAM:**

#include <iostream>

#include <stdlib.h>

#include <string>

using namespace std;

class Account

{

public:

Account(double a=0)

{

if (a < 0)

{

cout << "Entered Value Was Invalid, Hence it is Set to $0.0" << endl;

account\_Balance = 0;

}

else

account\_Balance = a;

}

void credit()

{

cout << "Your Current Balance before Crediting : $" << account\_Balance << endl;

double add;

cout << "Enter Amount to Credit into your Current Balance : $";

cin >> add;

if (add >= 0)

{

account\_Balance = account\_Balance + add;

}

else

cout << "Invalid Amount had Entered !" << endl;

cout << endl <<"Your Current Balance after Debbiting : $" << account\_Balance << endl;

}

void debit()

{

cout << "Your Current Balance before Debbiting : $" << account\_Balance << endl;

double subtract;

cout << "Enter Amount to Debit from your Current Balance : $";

cin >> subtract;

if (account\_Balance < subtract)

{

cout << " Debit Amount Exceeded Account Balance !" << endl;

account\_Balance = account\_Balance;

}

else

account\_Balance = account\_Balance - subtract;

cout << endl <<"Your Current Balance after Debbiting : $" << account\_Balance << endl;

}

void get\_Balance()

{

cout << "Your Current Balance is : $" << account\_Balance;

}

double account\_Balance;

};

class Saving\_Account:public Account

{

public:

Saving\_Account(int a = 0, int i = 0)

{

if (a < 0)

{

cout << "Entered Value Was Invalid, Hence it is Set to $0.0" << endl;

account\_Balance = 0;

}

else

account\_Balance = a;

Interest = i;

}

void calculate\_Interest()

{

cout << endl << "Account Balance before Adding Interest Rates : $" << account\_Balance << endl;

account\_Balance = account\_Balance + ((account\_Balance \* Interest)/100);

cout << "Account Balance after Adding Interest Rates : $" << account\_Balance << endl;

}

double Interest;

};

class Checking\_Account :public Account

{

public:

Checking\_Account(int a=0)

{

if (a < 0)

{

cout << "Entered Value Was Invalid, Hence it is Set to $0.0" << endl;

account\_Balance = 0;

}

else

account\_Balance = a;

fee\_Charged = 2;

}

void credit()

{

cout << "Your Current Balance before Crediting : $" << account\_Balance << endl;

double add;

cout << "Enter Amount to Credit into your Current Balance : $";

cin >> add;

if (add >= 0)

{

account\_Balance = account\_Balance + add - 2;

}

else

cout << "Invalid Amount had Entered !" << endl;

cout << "$2 Are Charged For Creditting ! " << endl;

cout << endl << "Your Current Balance after Debbiting : $" << account\_Balance << endl;

}

void debit()

{

cout << "Your Current Balance before Debbiting : $" << account\_Balance << endl;

double subtract;

cout << "Enter Amount to Debit from your Current Balance : $";

cin >> subtract;

if (account\_Balance < subtract)

{

cout << " Debit Amount Exceeded Account Balance !" << endl;

account\_Balance = account\_Balance;

}

else

account\_Balance = account\_Balance - subtract - 2;

cout << "$2 Are Charged For Debitting ! " << endl;

cout << endl << "Your Current Balance after Debbiting : $" << account\_Balance << endl;

}

double fee\_Charged;

};

int main()

{

Account A(30);

Saving\_Account S(40);

Checking\_Account C(50);

cout << "Account 1 :" << endl;

A.get\_Balance();

cout << endl << "Account 2 :" << endl; // displaying values

S.get\_Balance();

cout << endl << "Account 3 :" << endl;

C.get\_Balance();

cout << endl;

cout << endl << "Account 1 :" << endl;

A.credit();

cout << endl << "Account 2 :" << endl; // creditting values

S.credit();

cout << endl << "Account 3 :" << endl;

C.credit();

cout << endl;

cout << endl << "Account 1 :" << endl;

A.debit();

cout << endl << "Account 2 :" << endl; // debitting values

S.debit();

cout << endl << "Account 3 :" << endl;

C.debit();

cout << endl << endl;

system("pause");

}

**A screenshot of a computer screen

Description automatically generated**

**A screenshot of a computer screen

Description automatically generated**