Q1. Define a class Complex to represent a complex number with instance variables a and b to store real and imaginary parts.

Also define following member functions

a. void setData(int,int)

b. void showData()

c. Complex add(Complex)

#include<iostream>

using namespace std;

class Complex{

private:

int a, b;

public:

void setData(int r, int i){

a = r;

b = i;

}

void showData(void){

cout<<a<<"+"<<b<<"i";

}

Complex add(Complex t){

t.a = a + t.a;

t.b = b + t.b;

return t;

}

};

int main(){

Complex c1, c2, ans;

c1.setData(5,2);

c2.setData(1,3);

ans = c1.add(c2);

ans.showData();

}

Q2. Define a class Time to represent a time with instance variables h,m and s to store hour, minute and second.

Also define following member functions

a. void setTime(int,int,int)

b. void showTime()

c. void normalize()

d. Time add(Time)

#include <iostream>

using namespace std;

class Time {

private:

int h, m, s;

public:

void setTime(int, int, int);

void showTime(void);

void normalize(void);

Time add(Time);

};

int main() {

Time t;

int h, m, s;

cout<<"Enter Hour, Minutes, Seconds : ";

cin>>h>>m>>s;

t.setTime(h, m, s);

t.normalize();

cout<<"Before Add Function Call : ";

t.showTime();

cout<<"After Add Function Call : ";

t = t.add(t);

t.showTime();

}

void Time::setTime(int hr, int mi, int sec){

h = hr;

m = mi;

s = sec;

}

void Time::showTime(void){

cout<<h<<" : "<<m<<" : "<<s;

}

void Time::normalize(void){

if(s >= 60){

m += s / 60;

s %= 60;

}

if (m >= 60){

h += m / 60;

m %= 60;

}

}

Time Time::add(Time t){

Time tmp;

cout<<"Enter Hour, Minutes, Seconds : ";

cin>>tmp.h>>tmp.m>>tmp.s;

tmp.h += t.h;

tmp.m += t.m;

tmp.s += t.s;

tmp.normalize();

return tmp;

}

Q3. Define a class Cube and calculate Volume of Cube and initialise it using constructor.

#include <iostream>

using namespace std;

class Cube{

private:

float a;

public:

Cube(float x){

a = x;

}

void showData(void){

cout<< "Volume of cube : "<<a\*a\*a;

}

};

int main() {

Cube c(2);

c.showData();

}

Q4. Define a class Counter and Write a program to Show Counter using Constructor.

#include <iostream>

using namespace std;

class Counter{

public:

Counter(){

cout<<"Your are in Counter Class";

}

};

int main(){

Counter c;

}

Q5. Define a class Date and write a program to Display Date and initialise date object using Constructors.

#include <iostream>

using namespace std;

class Date{

private:

int d, m, y;

public:

Date(){

d = 13;

m = 03;

y = 2024;

}

void show(){

cout<<d<<"/"<<m<<"/"<<y;

}

};

int main(){

Date date;

date.show();

}

Q6. Define a class student and write a program to enter student details using constructor and define member function to display all the details

#include <iostream>

using namespace std;

class Student {

private:

char name[30], grade[6];

int roll;

public:

Student() {

cout << "Enter Student name :";

fgets(name, 30, stdin);

fflush(stdin);

cout << "Enter Grade :";

fgets(grade, 6, stdin);

fflush(stdin);

cout << "Enter Roll no. :";

cin >> roll;

}

void show(void) {

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

cout << "Grade :" << grade;

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

}

};

int main() {

Student s;

s.show();

}

Q7. Define a class Box and write a program to enter length, breadth and height and

initialize objects using constructor also define member functions to calculate volume of the box.

#include <iostream>

using namespace std;

class Box {

private:

float length, breadth, height;

public:

Box(int l, int b, int h) {

length = l;

breadth = b;

height = h;

}

void cal(void) {

cout << "Volume of the box : " << length \*breadth \*height;

}

};

int main() {

Box b(5, 5, 5);

b.cal();

}

Q8. Define a class Bank and define member functions to read principal , rate of interest and year. Another member functions to calculate simple interest and display it. Initialise all details using constructor.

#include <iostream>

using namespace std;

class Bank {

private:

float principal, year, roi;

public:

Bank(float p, float y, float r) {

principal = p;

year = y;

roi = r;

}

void read(void) {

cout << "Your Principal : " << principal << endl;

cout << "Your Rate of interest : " << roi << endl;

cout << "Your Duration : " << year << " year";

}

void cal(void) {

float si = (principal \* roi \* year) / 100;

cout << endl << endl << "Simple Interest : " << si;

}

};

int main() {

Bank b(10000, 10, 6);

b.read();

b.cal();

}

Q9.Define a class Bill and define its member function get() to take detail of customer ,calculateBill() function to calculate electricity bill using below tariff :

Upto 100 unit RS. 1.20 per unit

From 100 to 200 unit RS. 2 per unit

Above 200 units RS. 3 per unit.

#include <iostream>

using namespace std;

class Bill {

private:

char name[60];

float unit;

float amount;

public:

void get(void);

void calculateBill();

};

void Bill::get(void) {

cout << "Enter Customer Name : ";

fgets(name, 60, stdin);

cout << "Enter Unit : ";

cin >> unit;

}

void Bill::calculateBill() {

if (unit <= 100) {

amount = unit \* 1.20;

} else if (unit >= 100 && unit <= 200) {

amount = 99 \* 1.20;

unit = unit - 99;

amount = amount + (unit \* 2);

} else {

amount = 99 \* 1.20;

unit = unit - 99;

amount = amount + (101 \* 2);

unit = unit - 101;

amount = amount + (unit \* 3);

}

cout << endl << "Electricity Bill : " << amount << " Rs.";

}

int main() {

Bill b;

b.get();

b.calculateBill();

}

Q10. Define a class StaticCount and create a static variable. Increment this variable in a function and call this 3 times and display the result.

#include <iostream>

using namespace std;

class StaticCount {

public:

static int count;

void fun(void) {

count = count + 1;

}

void disp(void) {

cout << count;

}

};

int main() {

StaticCount c;

c.fun();

c.fun();

c.fun();

c.disp();

}