**SVKM’s NMIMS**

**Mukesh Patel School of Technology Management and Engineering, Mumbai**

**Department of Electronics & Telecommunication**



**Programming for Problem Solving (Exp 12)**

|  |  |
| --- | --- |
| **Roll No: J001** | **Name: Adith Ramakrishna** |
| **Program: B. Tech Data Science (1st)** | **Batch: J1** |
| **Date of Experiment: 30/11/2022** | **Date of Submission: 5/12/2022** |

**Task 1:**

**Code:**

#include <iostream>

using namespace std;

class Shapes {

private:

int Area;

public:

void calArea(float l, float w) {

cout << "Area of rectangle: " << l \* w << endl;

}

void calArea(float s) {

cout << "Area of square: " << s \* s << endl;

}

};

int main() {

Shapes s;

s.calArea(10.0, 15.0);

s.calArea(10.0);

return 0;

}

**Task 2:**

**Code:**

#include <iostream>

using namespace std;

class Time {

private:

int hours, mins;

public:

Time(int h = 0, int m = 0) {

if(m < 60) {

hours = h;

mins = m;

}

else {

mins = m % 60;

hours = h + m / 60;

}

}

Time operator + (Time const & obj) {

Time res;

res.hours = hours + obj.hours;

res.mins = mins + obj.mins;

if(res.mins >= 60) {

res.hours = res.hours + res.mins / 60;

res.mins = res.mins % 60;

}

return res;

}

void print() {

cout << hours << " hours and " << mins << " mins\n";

}

};

int main() {

int h1, m1, h2, m2;

cout << "Enter Time 1 (hours mins): ";

cin >> h1 >> m1;

Time t1(h1, m1);

cout << "Enter Time 2 (hours mins): ";

cin >> h2 >> m2;

Time t2(h2, m2);

cout << "\nSum: ";

(t1 + t2).print();

}

**Task 3:**

**Code:**

#include <iostream>

using namespace std;

class Distance {

private:

int feet, inches;

public:

Distance(int f = 0, int i = 0) {

if(i < 12) {

feet = f;

inches = i;

}

else {

inches = i % 12;

feet = f + i / 12;

}

}

Distance operator + (Distance const & obj) {

Distance res;

res.feet = feet + obj.feet;

res.inches = inches + obj.inches;

if(res.inches >= 12) {

res.feet = res.feet + res.inches / 12;

res.inches = res.inches % 12;

}

return res;

}

void print() {

cout << feet << " ft " << inches << " inches\n";

}

};

int main() {

int f1, i1, f2, i2;

cout << "Enter Distance 1 (feet inches): ";

cin >> f1 >> i1;

Distance d1(f1, i1);

cout << "Enter Distance 2 (feet inches): ";

cin >> f2 >> i2;

Distance d2(f2, i2);

cout << "\nSum: ";

(d1 + d2).print();

}

**Task 4:**

**Code:**

#include<iostream>

using namespace std;

int sum(int x, int y) {

return x + y;

}

double sum(double x, double y) {

return x + y;

}

int sum(int x, int y, int z) {

return x + y + z;

}

int main() {

cout << "The Sum of 2 Integers (10, 20): " << sum(10, 20) << endl;

cout << "The Sum of 3 Integers (10, 20, 30): " << sum(10, 20, 30) << endl;

cout << "The Sum of 2 Floats (10.6, 10.6): " << sum(10.6, 10.6) << endl;

}

**Homework Questions:**

**1:**

**Constructor overloading means having more than one constructor with the same name. Constructors are methods invoked when an object is created.**

**2:**

**Some operators cannot be overloaded which include sizeof operator, typeid, Scope resolution (::), Class member access operator (.), Ternary or conditional operator (?:).**