VIT - Vellore

Name: RONIT MEXSON .
Email: ronit.mexson2024@vitstudent.ac.in
Roll no: 24BAI0036
Phone: 9999999999
Branch: ARUMUGA ARUN R_OOPS
Department: admin
Batch: VL2024250502365
Degree: admin

Scan to verify results

BCSE102P_Structured and Object Oriented Programming Lab_VL2024250502365

VIT V_Structured and OOP_Lab 7_COD_Easy_Function Overloading

Attempt : 1
Total Mark : 20
Marks Obtained : 20

Section 1 : Coding

1. Problem Statement

Vamsi is a young and curious student who is eager to learn about finding the minimum number among a set of integers and double values. He needs a program to find the minimum number from a given set of values.

Help him solve the program by overloading the function named findMin.

Answer

```
// You are using GCC
#include <iostream>
#include <iomanip>
using namespace std;
```

```
int findMin(int a, int b, int c) {
    return min(a, min(b, c));
}


double findMin(double a, double b, double c) {
    return min(a, min(b, c));
}

int main() {
    int x, y, z;
    double p, q, r;


    cin >> x >> y >> z;

    cin >> p >> q >> r;

    // Output
    cout << "Minimum integer: " << findMin(x, y, z) << endl;
    cout << fixed << setprecision(2);
    cout << "Minimum double-point value: " << findMin(p, q, r) << endl;

    return 0;
}
```

Status : Correct **Marks : 10/10**

2. Problem Statement

Teju is working on a unit converter application that can convert lengths from millimetres to centimetres and kilometres. The application uses function overloading to handle the different unit types.

She wants to implement a UnitConverter class that contains two overloaded functions for conversion:

double convert(double millimetres) - If the unit type is 1, convert the value to centimetres.double convert(int unitType, double value) - If the unit type is 2, convert the value to kilometres.

Assist Teju in completing the converter application.

Answer

```
// You are using GCC
#include <iostream>
#include <iomanip>
using namespace std;

class UnitConverter {
public:
    double convert(double millimetres) {
        return millimetres / 10.0;
    }

    double convert(int unitType, double value) {
        if (unitType == 2)
            return value / 1000000.0;
        else
            return -1;
    }
};

int main() {
    int unitType;
    double value;

    cin >> unitType;

    if (unitType == 1 || unitType == 2) {
        cin >> value;
    } else {
        cout << "Invalid unit type!" << endl;
        return 0;
    }

    UnitConverter converter;
    cout << fixed << setprecision(2);

    if (unitType == 1) {
        cout << converter.convert(value) << " cm" << endl;
    } else if (unitType == 2) {
```

```
        cout << converter.convert(unitType, value) << " km" << endl;
    } else {
        cout << "Invalid unit type!" << endl;
    }

    return 0;
}
```

Status : Correct ***Marks : 10/10***