

## Functions and Classes

### Objectives

- Create functions using pass by value and pass by reference
- Create a simple class

### Programming Problem 1

Write a program that asks a user if he or she would like to convert from metric to US customary units or from the US system to metric. You will only need to work with length, so meters, feet, and inches. To implement this program, you will need to ask the user if he or she wants to convert from meters to feet and inches or from feet and inches to meters. Include in the program a loop that asks the user if he or she wants to do another conversion or to stop.

The functions you should write include:

```
void inputM ( double& meters );
//Prompt given to the user for input of a number of meters
//as a double

void inputUS ( int& feet, double& inches );
//Prompt given to the user for input in the format FF II,
//where FF is an int number of feet and II is a double number
//of inches feet and inches are returned as entered by the
//user.

void convertUStoM(int feet, double inches,
                 double& meters );
//REQUIRED CONSTANTS: INCHES_PER_FOOT, METERS_PER_FOOT
//inches < 12, feet within range of values for an int

//meters assigned 0.3048 * (feet + inches/12)
//Observe that the requirement to produce centimeters is met
//by the value of the first two decimal places of meters.

void convertMtoUS (int& feet, double& inches,
                  double meters );
// REQUIRED CONSTANTS: INCHES_PER_FOOT, METERS_PER_FOOT
//the variable feet is assigned the integer part of
//meters/METERS_PER_FOOT
//the variable inches is assigned the fractional part of
//feet after conversion to inch units.

void output( int feet, double inches, double meters );
```



```

//input: the formal argument for meters fits into a double
//output:
//"the value of feet, inches" <feet, inches>
//" corresponds to meters, centimeters is " <meters>
//where meters is displayed as a number with two decimal
//places

void USToMetric();
// requests US measure, converts to metric, outputs both

void MetricToUS();
// request metric measure, converts to US units, outputs both

```

To help you out, here is the main() to call your functions:

```

int main()
{
    char ans;
    do
    {
        int which;
        cout << "Enter 1 for US units to Metric or " << endl
        << "Enter 2 for Metric to US units conversion"
        << endl;
        cin >> which;
        if ( 1 == which )
            USToMetric();
        else
            MetricToUS();

        cout << "Y or y allows another choice of conversion. "
        << "any other quits" << endl;
        cin >> ans;
    } while ( 'y' == ans || 'Y' == ans );
    cout << "All done." << endl;
    return 0;
}

```

## Programming Problem 2

Define a class called Odometer that will be used to track fuel and mileage for an automobile. Include private member variables to track the miles driven and the fuel efficiency of the vehicle in miles per gallon. The class should also have a constructor that initializes these values to zero. Also include a member function to reset the odometer to zero miles, a member function to set the fuel efficiency, a member function that accepts miles driven for a trip and adds it to the odometer's total, and a member function that returns the number of gallons that the vehicle has consumed since the odometer was last reset.

You can use the main() below to test your program:

```
int main()
{
    // Two test trips
    Odometer trip1, trip2;

    trip1.reset();
    trip1.setFuelEfficiency(45);
    trip1.logMiles(100);
    cout << "For your fuel-efficient small car:" << endl;
    cout << "After 100 miles, " << trip1.gasConsumed() <<
    " gallons were used." << endl;
    trip1.logMiles(50);
    cout << "After another 50 miles, " << trip1.gasConsumed() <<
    " gallons were used." << endl;

    cout << endl;

    trip2.reset();
    trip2.setFuelEfficiency(13);
    trip2.logMiles(100);
    cout << "For your gas guzzler:" << endl;
    cout << "After 100 miles, " << trip2.gasConsumed() <<
    " gallons were used." << endl;
    trip2.logMiles(50);
    cout << "After another 50 miles, " << trip2.gasConsumed() <<
    " gallons were used." << endl;

    return 0;
}
```

**Please submit two files to Moodle:**

**conversion\_firstname\_lastname.cpp for Program 1**

**odometer\_firstname\_lastname.cpp for Program 2**