

Lab 11

LAB 11.1 Working with Basic Structures

Retrieve program `rect_struct.cpp` from the Lab 11 folder. The code is as follows:

```
#include <iostream>
#include <iomanip>
using namespace std;

// This program uses a structure to hold data about a rectangle

// PLACE YOUR NAME HERE

// Fill in code to declare a structure named rectangle which has
// members length, width, area, and perimeter all of which are floats


int main()
{
    // Fill in code to define a rectangle variable named box

    cout << "Enter the length of a rectangle: ";

    // Fill in code to read in the length member of box

    cout << "Enter the width of a rectangle: ";

    // Fill in code to read in the width member of box

    cout << endl << endl;

    // Fill in code to compute the area member of box
    // Fill in code to compute the perimeter member of box

    cout << fixed << showpoint << setprecision(2);

    // Fill in code to output the area with an appropriate message
    // Fill in code to output the perimeter with an appropriate message

    return 0;
}
```

Exercise 1

Fill in the code (places in bold) so that the program will compute and print out the area and perimeter of the rectangle based on the user's input length and width.

Exercise 2

Add code to the program so that the modified program will determine whether or not the rectangle entered by the user is a square.

Sample Run:

```
Enter the length of a rectangle: 7
Enter the width of a rectangle: 7
The area of the rectangle is 49.00
The perimeter of the rectangle is 28.00
The rectangle is a square.
```

LAB 11.2 Initializing Structures

Retrieve program `init_struct.cpp` from the Lab 11 folder. The code is as follows:

```
#include <iostream>
#include <string>
#include <iomanip>
using namespace std;

// This program demonstrates partially initialized structure variables

// PLACE YOUR NAME HERE

struct taxpayer
{
    string name;
    long socialSecNum;
    float taxRate;
    float income;
    float taxes;
};

int main()
{
    // Fill in code to initialize a structure variable named citizen1 so that
    // the first three members are initialized. Assume the name is Tim
    // McGuinness, the social security number is 255871234, and the tax rate is .35

    // Fill in code to initialize a structure variable named citizen2 so that
    // the first three members are initialized. Assume the name is John Kane,
    // the social security number is 278990582, and the tax rate is .29

    cout << fixed << showpoint << setprecision(2);

    // calculate taxes due for citizen1
    // Fill in code to prompt the user to enter this year's income for the citizen1

    // Fill in code to read in this income to the appropriate structure member

    // Fill in code to determine this year's taxes for citizen1

    cout << "Name: " << citizen1.name << endl;
    cout << "Social Security Number: " << citizen1.socialSecNum << endl;
    cout << "Taxes due for this year: $" << citizen1.taxes << endl << endl;

    // calculate taxes due for citizen2
    // Fill in code to prompt the user to enter this year's income for citizen2

    // Fill in code to read in this income to the appropriate structure member

    // Fill in code to determine this year's taxes for citizen2

    cout << "Name: " << citizen2.name << endl;
    cout << "Social Security Number: " << citizen2.socialSecNum << endl;
    cout << "Taxes due for this year: $" << citizen2.taxes << endl << endl;

    return 0;
}
```

Exercise 1

Fill in the code (places in bold) to make the program work as follows:

Sample Run:

Please input the yearly income for Tim McGuinness: 30000
Name: Tim McGuinness
Social Security Number: 255871234
Taxes due for this year: \$10500.00

Please input the yearly income for John Kane: 60000
Name: John Kane
Social Security Number: 278990582
Taxes due for this year: \$17400.00

LAB 11.3 Arrays of Structures

Retrieve program `array_struct.cpp` from the Lab 11 folder. The code is as follows:

```
#include <iostream>
#include <iomanip>
using namespace std;

// This program demonstrates how to use an array of structures

// PLACE YOUR NAME HERE

// Fill in code to declare a structure called taxPayer that has three
// members: taxRate, income, and taxes - each of type float

int main()
{
    // Fill in code to define an array named citizen which holds
    // 5 taxPayers structures

    cout << fixed << showpoint << setprecision(2);
    cout << "Please enter the annual income and tax rate for 5 tax payers: ";
    cout << endl << endl << endl;

    for(int count = 0; count < 5; count++)
    {
        cout << "Enter this year's income for tax payer " << (count + 1);
        cout << ": ";

        // Fill in code to read in the income to the appropriate place

        cout << "Enter the tax rate for tax payer # " << (count + 1);
        cout << ": ";

        // Fill in code to read in the tax rate to the appropriate place

        // Fill in code to compute the taxes for the citizen and store it
        // in the appropriate place

        cout << endl;
    }

    cout << "Taxes due for this year: " << endl << endl;

    // Fill in code for the first line of a loop that will output the
    // tax information

    {
        cout << "Tax Payer # " << (index + 1) << ": " << "$ "
            << citizen[index].taxes << endl;
    }

    return 0;
}
```

Exercise 1

Fill in the code (places in bold) so that the program is able to read in 5 taxpayer's income and tax rate. The program should also output the tax information of all 5 taxpayers.

Sample Run:

```
Enter this year's income for tax payer 1: 45000
Enter the tax rate for tax payer # 1: .19
Enter this year's income for tax payer 1: 60000
Enter the tax rate for tax payer # 1: .23
Enter this year's income for tax payer 1: 12000
Enter the tax rate for tax payer # 1: .01
Enter this year's income for tax payer 1: 104000
Enter the tax rate for tax payer # 1: .30
Enter this year's income for tax payer 1: 50000
Enter the tax rate for tax payer # 1: .22
```

```
Tax Payer # 1: $ 8550.00
Tax Payer # 2: $ 13800.00
Tax Payer # 3: $ 120.00
Tax Payer # 4: $ 31200.00
Tax Payer # 5: $ 11000.00
```

Exercise 2

In the previous code we have the following:

```
cout << "Tax Payer # " << (index+1) << ": " << "$ " << citizen[index].taxes << endl;
```

Why do you think we need (index+1) in the first line but index in the second line?

LAB 11.4 Student Generated Code Assignments

Option 1: Re-write your final program in Lab 11.2 so that it uses an array of structures. Write the program so that the array will store many `taxPayer` structures. You may choose to maintain the same program output as before, or you may opt to modify the program for better usability (e.g. allow user to continue entering new taxpayer entries, with a sentinel value to stop).