

Tutorial 1 - Basic C++ Programming

1. **(Reference)** Assume the following declaration:

```
int num1 = 100;  
int num2 = 200;  
int *p = &num1;  
int &ref = *p;
```

We also assume that the address of num1 is 8800 and the address of num2 is 9976.

That is,

8800	<input type="text"/>	num1
	...	
	...	
	...	
9976	<input type="text"/>	num2

Determine the value of the following variables:

(a) num1 (b) num2 (c) p (d) *p (e) ref (f) &ref

when we execute the following lines of code. Note all of the results are cumulative.

```
(i) *p = 50;  
(ii) ref = ref / 2;  
(iii) p = &num2; *p = 400;  
(iv) ref = num2; ref = ref * 2;  
(v) ref = &num2;
```

2. **(calArea1)** Write three overloaded functions `calArea()` in C++ to calculate the area of a square, a rectangle and a trapezoid. The detailed math equations are as follows.

- a) The area of a square:

$$A = s^2$$

where s is the side length of the square.

- b) The area of a rectangle:

$$A = l \times w$$

where l is the length of the rectangle and w is the width of the rectangle.

- c) The area of a trapezoid:

$$A = \frac{1}{2}(b_1 + b_2)h$$

where b_1 and b_2 are the top and bottom base lengths, and h is the height.

Suppose that all the function parameters of `calArea()` are **integers**. Below is the main function for you to test your implementations. **Note:** the main function below assumes that the users will NOT input an invalid type of data.

```

#include <iostream>
using namespace std;
// T0-D0: Write your functions here

int main() {
    int choice;

    while (true) {
        // Display menu options
        cout << "\nChoose an option:\n";
        cout << "1 - Square\n";
        cout << "2 - Rectangle\n";
        cout << "3 - Trapezoid\n";
        cout << "0 other - Exit\n";
        cout << "Enter your choice (int): ";
        cin >> choice;

        if (choice == 1) {
            // Square
            int side;
            cout << "Enter the side length of the square (int): ";
            cin >> side;
            cout << "Area of Square: " << calArea(side) << endl;
        }
        else if (choice == 2) {
            // Rectangle
            int length, width;
            cout << "Enter the length and width of the rectangle (int): ";
            cin >> length >> width;
            cout << "Area of Rectangle: " << calArea(length, width) << endl;
        }
        else if (choice == 3) {
            // Trapezoid
            int base1, base2, height;
            cout << "Enter the two bases and height of the trapezoid (int): ";
            cin >> base1 >> base2 >> height;
            cout << "Area of Trapezoid: " << calArea(base1, base2, height) << endl;
        }
        else {
            // Exit program
            cout << "Exiting program..." << endl;
            break;
        }
    }

    return 0;
}

```

Here are sample test cases:

Choose an option:

1 - Square

2 - Rectangle

3 - Trapezoid

Other - Exit

Enter your choice (int): 2

Enter the length and width of the rectangle (int): 10 20

Area of Rectangle: 200

Choose an option:

1 - Square

2 - Rectangle

3 - Trapezoid

Other - Exit

Enter your choice (int): 3

Enter the two bases and height of the trapezoid (int): 20 40 2

Area of Trapezoid: 60

Choose an option:

1 - Square

2 - Rectangle

3 - Trapezoid

Other - Exit

Enter your choice (int): 1

Enter the side length of the square (int): 20

Area of Square: 400

Choose an option:

1 - Square

2 - Rectangle

3 - Trapezoid

Other - Exit

Enter your choice (int): 0

Exiting program...

3. (**calArea2**) Question 2 assumes that the input of `calArea()` can only be integers, which is actually not true in real world. You are asked write function templates to calculate the area of squares, rectangles and trapezoids, where the input parameter type can be any types of numbers, e.g., int, double, float, long.

Below is the main function for you to test your implementations.

```
#include <iostream>
using namespace std;
// T0-D0: Write Your Code Here
```

```

int main() {
    // Test cases
    int side1 = 5;
    cout << "Area of Square: " << calArea(side1) << endl;
    double side2 = 11.11;
    cout << "Area of Square: " << calArea(side2) << endl;

    int length1 = 10, width1 = 20;
    cout << "Area of Rectangle: " << calArea(length1, width1) << endl;
    float length2 = 23.4, width2 = 10.8;
    cout << "Area of Rectangle: " << calArea(length2, width2) << endl;

    long b1 = 20, b2 = 40, height = 10;
    cout << "Area of Trapezoid: " << calArea(b1, b2, height) << endl;

    return 0;
}

```

4. **(studentInfo)** Write a C++ program to allow users to input a sequence of students' information such as name, grade or mark of a course, and finally calculate the average mark of all these students. Below are a few key points for you to follow in your program:
- a) The course result of a student can be either a grade ranging from "A", "B" to "C", or a mark ranging from 0 to 100. You are required to use union to implement it:

```

union Result {
    int mark;
    char grade; // Can be only "A", "B" and "C"
};

```

- b) Each student's information should be represented as a structure in C++, and include all the necessary information. The structure should have a function for converting the grade to finalMark, and also have a boolean variable to indicate if the course result input by users is grade or not. The rule to convert grades to marks is shown in the right table. If the course result is already a mark, then simply assign it to finalMark. The definition of the structure is as follows:

Grade	Final Mark
A	90
B	80
C	60

```

struct Student {
    char studentName[50];
    bool isGrade;
    int finalMark; //used to store the final mark

    // Union to store either mark or grade
    Result res;

    void convertGrade() { // A=90 , B=80, C=60
        // Please write your code here.
    }
}

```

```
};
```

- c) Different users may want to input the information of different number of students, so you should **use dynamic memory allocation** in your program. Your program should use `cin` and `cout` for user input and output. For simplicity, you can assume that the users **will NOT input an invalid type of data** for the requested user inputs.
- d) The function `displayStudentInfo()` displays each student's name and final mark, calculates the average mark of all the input users and displays it. The function prototype is as follows, where `students` is a pointer of `Student` pointing to an array of `Student` that stores students' information and `count` is the size of this array:
- ```
void displayStudentInfo(Student *students, int count);
```

The initial code skeleton is given below:

```
#include <iostream>
using namespace std;
union Result {
 int mark;
 char grade; // Can be only 'A', 'B' or 'C'
};

struct Student {
 char studentName[50];
 bool isGrade;
 int finalMark; // Used to store the final mark
 Result res;

 void convertGrade() {
 // To-do: WRITE YOUR CODE HERE
 }
};

void displayStudentInfo(Student *students, int count);

int main() {
 // To-do: WRITE YOUR CODE HERE

 return 0;
}
```

Sample test cases are given below.

### **Test Case 1**

```
How many students do you want to input?
Enter student size: 3
```

```
Enter student name: Harry Wang
Enter 'G' if result is grade or 'M' if result is mark: G
Enter grade (A,B,C): A
Enter student name: Hermione Heer
Enter 'G' if result is grade or 'M' if result is mark: M
Enter mark (0-100): 100
Enter student name: Ronald Cheung
Enter 'G' if result is grade or 'M' if result is mark: G
Enter grade (A,B,C): C

Student Results:
Name: Harry Wang, Final Mark: 90
Name: Hermione Heer, Final Mark: 100
Name: Ronald Cheung, Final Mark: 60

Average Final Mark: 83.3333
```

### **Test Case 2**

```
How many students do you want to input?
Enter student size: 5
Enter student name: Draco Lee
Enter 'G' if result is grade or 'M' if result is mark: G
Enter grade (A,B,C): B
Enter student name: Jeffrey Heer
Enter 'G' if result is grade or 'M' if result is mark: M
Enter mark (0-100): 35
Enter student name: Blaise Muzner
Enter 'G' if result is grade or 'M' if result is mark: M
Enter mark (0-100): 42
Enter student name: Gregory Bush
Enter 'G' if result is grade or 'M' if result is mark: M
Enter mark (0-100): 20
Enter student name: Vincent Leaung
Enter 'G' if result is grade or 'M' if result is mark: G
Enter grade (A,B,C): C

Student Results:
Name: Draco Lee, Final Mark: 80
Name: Jeffrey Heer, Final Mark: 35
Name: Blaise Muzner, Final Mark: 42
Name: Gregory Bush, Final Mark: 20
Name: Vincent Leaung, Final Mark: 60

Average Final Mark: 47.4
```