

Lab Session 5

Lab 5.1

[Skip Exercise 1 – Exercise 3]:

(3 points) Exercise 4: Retrieve and complete `sentinel.cpp` according to the instructions. Attach the completed `sentinel.cpp` program.

```
int main() {  
  
    int month = 1;  
    float total = 0, rain;  
  
    cout << "Enter the total rainfall for month " << month << endl;  
    cout << "Enter -1 when you are finished" << endl;  
    cin >> rain;  
  
    while (rain != -1) {  
        total = total + rain;  
        month++;  
  
        cout << "Enter the total rainfall in inches for month " << month << endl;  
        cout << "Enter -1 when you are finished " << endl;  
        cin >> rain;  
    }  
    if (month == 1)  
        cout << "No data has been entered" << endl;  
    else  
        cout << "The total rainfall for the " << month - 1 << " month is " << total << " inches." << endl;  
    system("pause>0");  
    return 0;  
}
```

```
E:\Study Materials\C++\Kira\Debug\Kira.exe  
Enter the total rainfall for month 1  
Enter -1 when you are finished  
5  
Enter the total rainfall in inches for month 2  
Enter -1 when you are finished  
3  
Enter the total rainfall in inches for month 3  
Enter -1 when you are finished  
2  
Enter the total rainfall in inches for month 4  
Enter -1 when you are finished  
-1  
The total rainfall for the 3 month is 10 inches.
```

(1 point) Exercise 5: Record the inputs you used, and the total rainfall you get.

Input	Input	Input
5	1	3
3	2	5
2	3	6
Total:	4	7
3 months 10	5	9
	6	2
	Total:	1
	6 months 21	2
		5
		Total:
	9 months 40	

(2 points) What happens if you enter -1 first?

It will generate an output “No data has been entered”.

(2 points) Is there any numerical data you should not enter?

Except from -1 we can input any numbers its because -1 is not equal to rain it will generate another statement.

(2 points) Exercise 6 (Extra credit):

What is the purpose of the following code in the program?

```
if (month == 1)
    cout << "No data has been entered" << endl;
```

If User does not give any data just input -1 in first input then this condition will be executed.

Lab 5.2

(1 point) Exercise 1: Retrieve and complete `dowhile.cpp`. Record below the cost calculated for the following combination of menu choices and number of cups:

Menu	Cups	Cost
Coffee	2	2
Tea	4	3
Hot Chocolate	5	6.25
Cappuccino	3	7.5
Exit	0	0

(3 points) Attach the modified `dowhile.cpp` program.

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

int main() {
    int number;
    float cost;
    char beverage;
    bool validBeverage;

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

    do
    {
        cout << endl << endl;
        cout << "Hot Beverage Menu" << endl << endl;
        cout << "A: Coffee          $1.00" << endl;
        cout << "B: Tea              $ .75" << endl;
        cout << "C: Hot chocolate    $1.25" << endl;
        cout << "D: Cappuccino       $2.50" << endl << endl << endl;

        cout << "Enter the beverage A,B,C or D you desire " << endl;
        cout << "Enter E to exit the program" << endl << endl;

        cin >> beverage;
        switch (beverage)
        {
            case 'a':
            case 'A':
            case 'b':
            case 'B':
            case 'c':
            case 'C':
            case 'd':
            case 'D': validBeverage = true;
                    break;
            default: validBeverage = false;
        }

        if (validBeverage == true)
        {
            cout << "How many cups would you like? " << endl;
            cin >> number;
        }
    }
```

```

switch (beverage)
{
case 'a':
case 'A': cost = number * 1.0;
cout << "The total cost is $ " << cost << endl;
break;
case 'b':
case 'B': cost = number * 0.75;
cout << "The total cost is $ " << cost << endl;
break;
case 'c':
case 'C': cost = number * 1.25;
cout << "The total cost is $ " << cost << endl;
break;
case 'd':
case 'D': cost = number * 2.50;
cout << "The total cost is $ " << cost << endl;
break;
case 'e':
case 'E': cout << " Please come again" << endl;
break;
default: cout << "Invalid character entered" << endl;
cout << "Try again please" << endl;
}
} while (beverage != 'e' && beverage != 'E');
system("pause>0");
return 0;
}

```

[Skip Exercise 2]

(2 points) Exercise 3: Replace `if (validBeverage == true)` with `if (validBeverage)` in `dowhile.cpp` and run the program again. Are there differences in the execution of the program? Why or why not?

No, because it has some Boolean value for both conditions.

Lab 5.3

Exercise 1: Open and examine `for.cpp` and answer the following questions:

(2 points) Why is the typecast operator needed to compute the mean in this statement:

```
mean = float(total) / value;?
```

Its because the total value is equal to zero so we have changed the int datatype to float with the help of type casting so we can get valid value.

(1 point) Replace the statement `mean = float(total) / value;` with `mean = total / value;` and run the program with the following values of `n`. Record the mean calculated by the modified program below:

Value	Mean (without typecasting)
4	2
5	3
6	3
7	4
8	4
9	5

(2 points) Are these results correct, and why/why not?

No because total and value variable both defined as int and their calculations also give int answer. We can use typecast a single variable to get answers in float

Restore the typecast operator to the statement that calculates mean

[Skip Exercise 2]

(3 points) Exercise 3: Modify `for.cpp` so that it computes the mean of consecutive positive integers $n, n+1, n+2, \dots, m$, where the user chooses n and m . For example, if a user picks $n = 3$, and $m = 9$, the program should find the mean of 3, 4, 5, 6, 7, 8 and 9, which is 6. The modified code must make sure that the user entered a first number n that is greater than zero, and that the user entered a second number m that is greater than the first number n . Attach the modified `for.cpp` program.

```
#include <iostream>
using namespace std;
int main()
{
    int value, last_value;
    int total = 0, counter = 0;
    int number;
    float mean;

    cout << "Please enter first n integer of series" << endl;
    cin >> value;

    cout << "Please enter last m integer of series " << endl;
    cin >> last_value;

    if (value > 0 && last_value > value)
    {
        for (number = value; number <= last_value; number++)
        {
            total = total + number;
            counter++;
        }

        cout << total << endl;
        mean = float(total) / (counter);

        cout << "The mean average of these " << counter
        << "positive integer is " << mean << endl;
    }
    else
        cout << "Invalid input - integer must be positive and last integer m should greater than first num
        system("pause>0");

    return 0;
}
```

```
E:\Study Materials\C++\Kira\64\Debug\Kira.exe
Please enter first n integer of series
2
Please enter last m integer of series
6
20
The mean average of these 5 positive integer is 4
```

Lab 5.4

(3 points) Exercise 1: Retrieve `nested.cpp` and modify the program so it lets the user decide how many days in a weekend a student works. Attach the modified `nested.cpp` program.

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

int main() {
    int numStudents;
    float numHours, total, average;
    int student, day = 0;
    int n;

    cout << "This program will find the average number of hours a day"
         << " that a student spent programming over a long weekend\n\n";
    cout << "How many students are there ?" << endl << endl;
    cin >> numStudents;
    cout << "Enter number of days for long weekend" << endl;
    cin >> n;

    for (student = 1; student <= numStudents; student++)
    {
        total = 0;
        for(day = 1; day <= n; day++)
        {
            cout << "Please enter the number of hours worked by student"
                 << student << " on day " << day << "." << endl;
            cin >> numHours;
            total = total + numHours;
        }
        average = total / n;
        cout << endl;
        cout << "The average number of hours per day spent programming by "
             << "student " << student << " is " << average
             << endl << endl << endl;
    }

    system("pause>0");
    return 0;
}
```

Select E:\Study Materials\C++\Kira\64\Debug\Kira.exe

This program will find the average number of hours a day that a student spent programming over a long weekend

How many students are there ?

2

Enter number of days for long weekend

2

Please enter the number of hours worked by student1 on day 1.

4

Please enter the number of hours worked by student1 on day 2.

6

The average number of hours per day spent programming by student 1 is 5

Please enter the number of hours worked by student2 on day 1.

9

Please enter the number of hours worked by student2 on day 2.

13

The average number of hours per day spent programming by student 2 is 11