**21/11/2020** **Mohd. Aftab Alam**
**02013302717**

# EXPERIMENT 11

**Aim : (a)** Write a Program in C/C++ to compute total salary of an employee when his basic salary is given. (Given: HRA = 3% of basic, DA = 8% of basic, CCA/MA = Rs. 100, Tax = Rs. 300, PF = Rs.780, TA = Rs. 800). Perform Slice based testing for all variables.

***Total Salary = (Basic + HRA + DA + TA) – (Tax + CM + PF)***

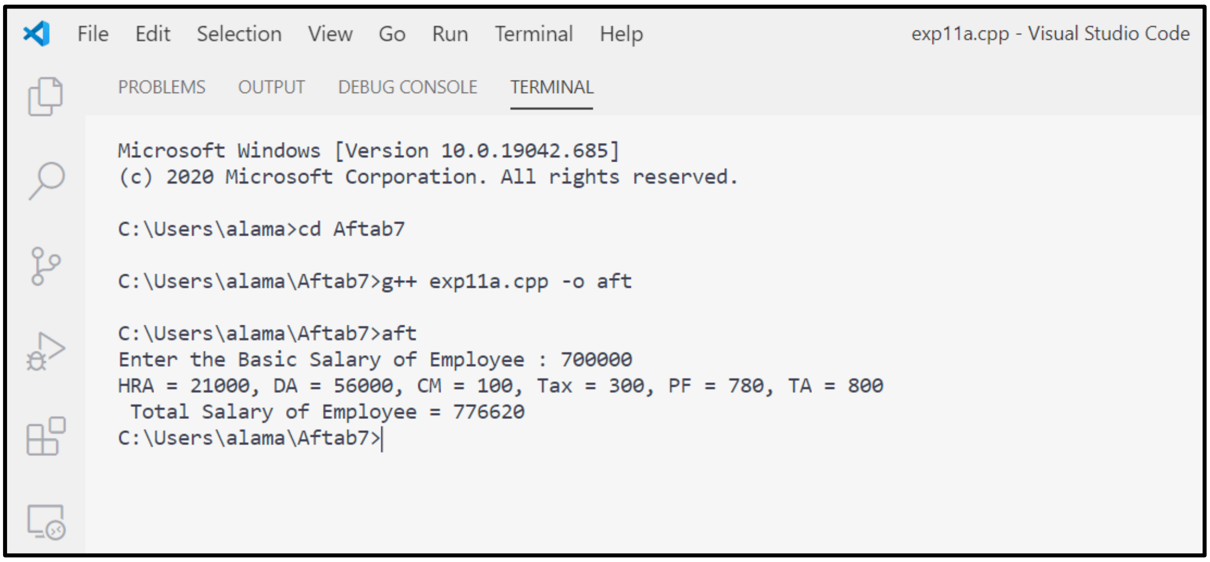
## Algorithm :

- Take the Basic Salary of the employee as input from the user.
- Calculate HRA and DA using the basic salary.
- Calculate the Total Salary by combining all the values.
- Print the Total Salary of the employee as calculated on the screen.

## Code :

```
#include <iostream>
using namespace std;
int main() {
float basic, HRA, DA, CM, tax, PF, TA, total_salary;
printf(“Enter the Basic Salary of Employee : “);
cin >> basic;
HRA = (basic * 3) / 100;
DA = (basic * 8) / 100;
CM = 100;
tax = 300;
PF = 780;
TA = 800;
cout<<”HRA=”<<HRA<<”DA=”<<DA<<”CM=”<<CM<<”Tax=”<<tax<<”PF=”<<PF<<”TA=”<<TA;
total_salary = (basic + HRA + DA + TA) – (tax + CM + PF);
cout << “\n Total Salary of Employee = “ << total_salary;
return 0;
}
```

## Output Screenshot :



```
Microsoft Windows [Version 10.0.19042.685]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\alama>cd Aftab7

C:\Users\alama\Aftab7>g++ exp11a.cpp -o aft

C:\Users\alama\Aftab7>aft
Enter the Basic Salary of Employee : 700000
HRA = 21000, DA = 56000, CM = 100, Tax = 300, PF = 780, TA = 800
 Total Salary of Employee = 776620
C:\Users\alama\Aftab7>
```

## Slice Based Testing :

There is total 8 variables in the program. We can create slices for each of them.

- ***Variable: basic***

**S(basic,5) / S(basic,15) = {1-5,15}**

```
1. int main() {
2. float basic, HRA, DA, CM, tax, PF, TA, total_salary;
3. printf("Enter the Basic Salary of Employee : ");
4. cin >> basic;
15.}
```

- ***Variable: HRA***

**S(HRA,6) / S(HRA,15) = {1-6,15}**

```
1. int main() {
2. float basic, HRA, DA, CM, tax, PF, TA, total_salary;
3. printf("Enter the Basic Salary of Employee : ");
4. cin >> basic;
5. HRA = (basic * 3) / 100;
15.}
```

- ***Variable: DA***

**S(DA,6) / S(DA,15) = {1-6,7,15}**

```
1. int main() {
2. float basic, HRA, DA, CM, tax, PF, TA, total_salary;
3. printf("Enter the Basic Salary of Employee : ");
4. cin >> basic;
6. DA = (basic * 8) / 100;
15.}
```

- ***Variable: CM***

**S(CM,8) / S(CM,15) = {1-3,8,15}**

```
1. int main() {
2. float basic, HRA, DA, CM, tax, PF, TA, total_salary;
7. CM = 100;
15.}
```

- ***Variable: tax***

**S(tax,8) / S(tax,15) = {1-3,8,15}**

```
1. int main() {
2. float basic, HRA, DA, CM, tax, PF, TA, total_salary;
8. tax = 300;
15.}
```

- ***Variable: PF***

**S(PF,10) / S(PF,15) = {1-3,10,15}**

```
1. int main() {
2. float basic, HRA, DA, CM, tax, PF, TA, total_salary;
9. PF = 780;
15.}
```

- ***Variable: TA***

**S(TA,11) / S(TA,15) = {1-3,11,15}**

```
1. int main() {
2. float basic, HRA, DA, CM, tax, PF, TA, total_salary;
10.TA = 800;
15.}
```

- ***Variable: total_salary***

**S(Total,12) = {1-12,15}**

```
1. int main() {
2. float basic, HRA, DA, CM, tax, PF, TA, total_salary;
3. printf("Enter the Basic Salary of Employee : ");
4. cin >> basic;
5. HRA = (basic * 3) / 100;
6. DA = (basic * 8) / 100;
7. CM = 100;
8. tax = 300;
9. PF = 780;
10.TA = 800;
12.total_salary = (basic + HRA + DA + TA) - (tax + CM + PF);
15.}
```

**S(Total,13) / S(Total,15) = {1-13,14,15}**

```
1. int main() {
2. float basic, HRA, DA, CM, tax, PF, TA, total_salary;
3. printf(“Enter the Basic Salary of Employee : “);
4. cin >> basic;
5. HRA = (basic * 3) / 100;
6. DA = (basic * 8) / 100;
7. CM = 100;
8. tax = 300;
9. PF = 780;
10. TA = 800;
11.cout<<”HRA=”<<HRA<<”DA=”<<DA<<”CM=”<<CM<<”Tax=”<<tax<<”PF=”<<PF<<”TA=”<<TA;
12. total_salary = (basic + HRA + DA + TA) – (tax + CM + PF);
13. cout << “\n Total Salary of Employee = “ << total_salary;
14. return 0;
15. }
```

**Test Cases :**

| S. No. | Slice | Lines Covered | Variables | | | Expected Output |
|---|---|---|---|---|---|---|
| | | | **Basic** | **HRA** | **DA** | |
| 1 | S(Basic,5) / S(Basic,15) | 1-5,15 | 1000 | 30 | 80 | No Output |
| 2 | S(HRA,6) / S(HRA,15) | 1-6,15 | 3000 | 90 | 240 | No Output |
| 3 | S(DA,7) / S(DA,15) | 1-5,7,15 | 3000 | 90 | 240 | No Output |
| 4 | S(MA,8) / S(MA,15) | 1-3,8,15 | 3000 | 90 | 240 | No Output |
| 5 | S(ITAX,9) / S(ITAX,15) | 1-3,9,15 | 3000 | 90 | 240 | No Output |
| 6 | S(PF,10) / S(PF,15) | 1-3,10,15 | 3000 | 90 | 240 | No Output |
| 7 | S(TA,11) / S(TA,15) | 1-3,11,15 | 3000 | 90 | 240 | No Output |
| 8 | S(Total,12) | 1-12,15 | 3000 | 90 | 240 | No Output |
| 9 | S(Total,13) / S(Total,15) | 1-13,14,15 | 5000 | 150 | 400 | 7530 |

**21/11/2020** **Mohd. Aftab Alam**
**02013302717**

**Aim :**

**(b)** Write a Program in C/C++ to find whether a triangle is right, acute or obtuse angled. Its input is a triplet of 3 positive integers (say a, b, c) from interval (1 to 100). Perform slice-based testing for all variables.

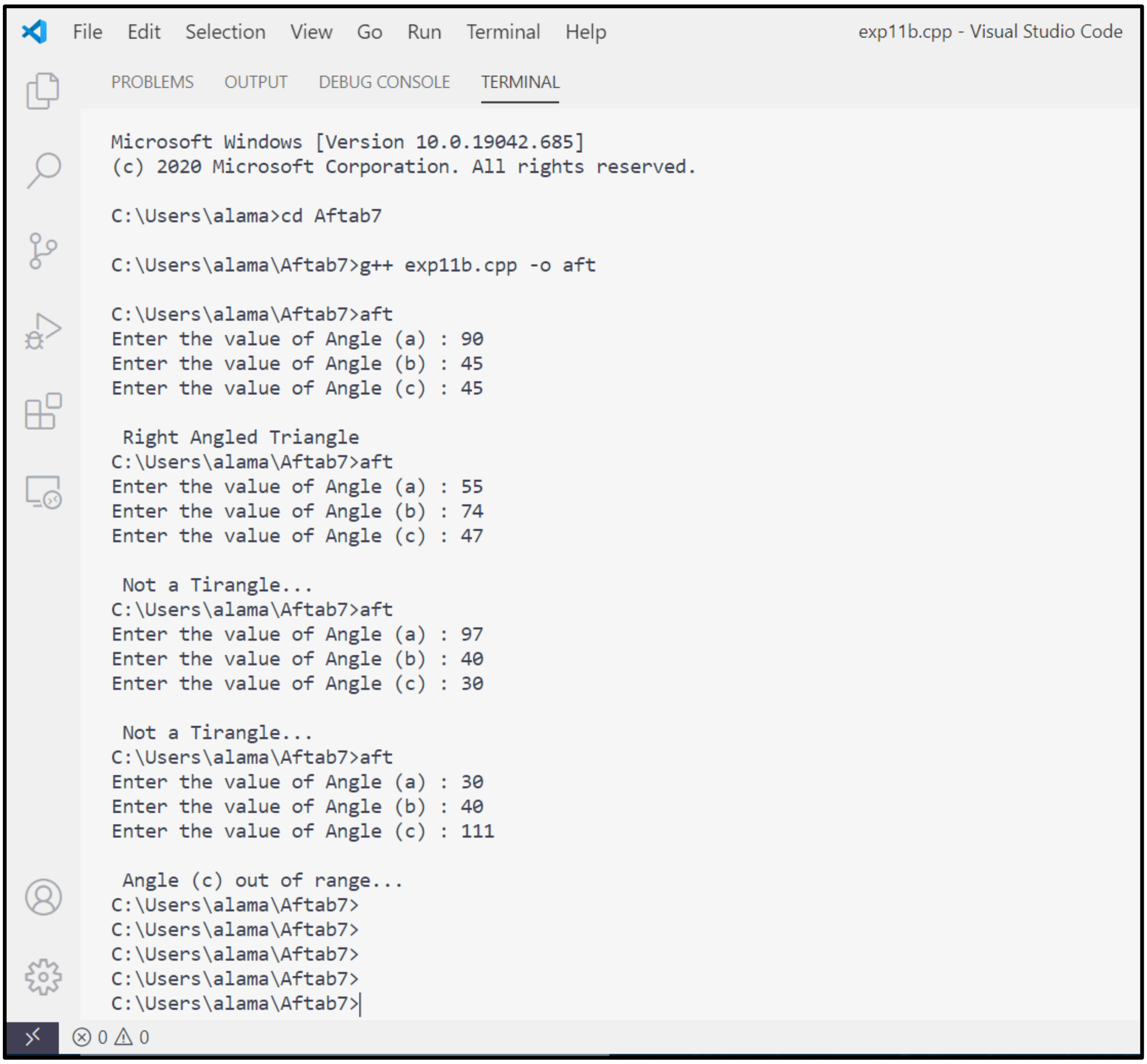
**Algorithm :**

- Take three input from the user for the angles a, b and c.
- Check whether they lie in the given interval.
- Check the sum of all the angles and if that is equal to 180° then proceed further, else it is not a triangle.
- After checking the validity of triangle check for following conditions and classify them into categories:
    - If any angle is 90°, then it is a Right-Angled Triangle.
    - If any angle is greater than 90°, then it is Obtuse Angled Triangle.
    - If any angle is smaller than 90°, then it is Acute Angled Triangle.

**Code:**

```
#include <iostream>
using namespace std;
int main() {
    int a, b, c, sum;
    cout << "Enter the value of Angle (a) : ";
    cin >> a;
    cout << "Enter the value of Angle (b) : ";
    cin >> b;
    cout << "Enter the value of Angle (c) : ";
    cin >> c;
    if (a > -1 && a < 101) {
        if (b > -1 && b < 101) {
            if (c > -1 && c < 101) {
                sum = a + b + c;
                if (sum == 180) {
                    if (a == 90 || b == 90 || c == 90) {
                        cout << "\n Right Angled Triangle";
                    }
                    else if (a > 90 || b > 90 || c > 90) {
                        cout << "\n Obtuse Angled Triangle";
                    }
                    else if (a < 90 && b < 90 && c < 90) {
                        cout << "\n Acute Angled Triangle";
                    }
                }
                else {
                    cout << "\n Not a Tirangle...";
                }
            }
            else {
                cout << "\n Angle (c) out of range...";
            }
        }
        else {
            cout << "\n Angle (b) out of range...";
        }
    }
    else
    {
        cout << "\n Angle (a) out of range...";
    }
    return 0;
}
```

**Output Screenshot :**



```
Microsoft Windows [Version 10.0.19042.685]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\alama>cd Aftab7

C:\Users\alama\Aftab7>g++ exp11b.cpp -o aft

C:\Users\alama\Aftab7>aft
Enter the value of Angle (a) : 90
Enter the value of Angle (b) : 45
Enter the value of Angle (c) : 45

 Right Angled Triangle
C:\Users\alama\Aftab7>aft
Enter the value of Angle (a) : 55
Enter the value of Angle (b) : 74
Enter the value of Angle (c) : 47

 Not a Tirangle...
C:\Users\alama\Aftab7>aft
Enter the value of Angle (a) : 97
Enter the value of Angle (b) : 40
Enter the value of Angle (c) : 30

 Not a Tirangle...
C:\Users\alama\Aftab7>aft
Enter the value of Angle (a) : 30
Enter the value of Angle (b) : 40
Enter the value of Angle (c) : 111

 Angle (c) out of range...
C:\Users\alama\Aftab7>
C:\Users\alama\Aftab7>
C:\Users\alama\Aftab7>
C:\Users\alama\Aftab7>
C:\Users\alama\Aftab7>
```

**Slice Based Testing :**

There is total 4 variables in the program. We can create slices for each of them.

- ***Variable: a***

**S (a,4) = {1,2,3,4,43}**

- ***Variable: b***

**S (b,4) = {1,2,3,4,43}**

- ***Variable: c***

**S (c,4) = {1,2,3,4,43}**

- ***Variable: sum***

**S (sum,4) = {1,2,3,4,43}**

**S (sum,4) = {1,2,3,4,5,6,7,8,9,10,11,12,13,14,29,34,38,41,43}**