**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;

**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;

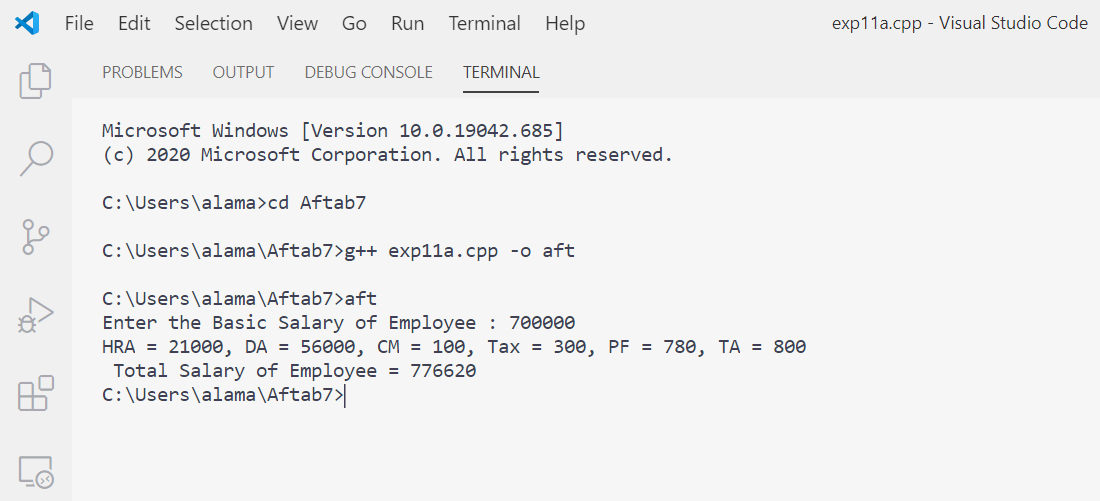
**12.** total\_salary = (basic + HRA + DA + TA) – (tax + CM + PF);

**13.** cout << “\n Total Salary of Employee = “ << total\_salary;

**14.** return 0;

**15.** }

**Output Screenshot :**



**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 |

**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:**

**1.** #include <iostream>

**2.** using namespace std;

**3.** int main() {

**4.**     int a, b, c, sum;

**5.**     cout << "Enter the value of Angle (a) : ";

**6.**     cin >> a;

**7.**     cout << "Enter the value of Angle (b) : ";

**8.**     cin >> b;

**9.**     cout << "Enter the value of Angle (c) : ";

**10.**    cin >> c;

**11.**    if (a > -1 && a < 101) {

**12.**        if (b > -1 && b < 101) {

**13.**            if (c > -1 && c < 101) {

**14.**                sum = a + b + c;

**15.**                if (sum == 180) {

**16.**                    if (a == 90 || b == 90 || c == 90) {

**17.**                        cout << "\n Right Angled Triangle";

**18.**                    }

**19.**                    else if (a > 90 || b > 90 || c > 90) {

**20.**                        cout << "\n Obtuse Angled Triangle";

**21.**                    }

**22.**                    else if (a < 90 && b < 90 && c < 90) {

**23.**                        cout << "\n Acute Angled Triangle";

**24.**                    }

**25.**                }

**26.**                else {

**27.**                    cout << "\n Not a Tirangle...";

**28.**                }

**29.**            }

**30.**            else {

**31.**                cout << "\n Angle (c) out of range...";

**32.**            }

**33.**        }

**34.**        else {

**35.**            cout << "\n Angle (b) out of range...";

**36.**        }

**37.**    }

**38.**    else

**39.**    {

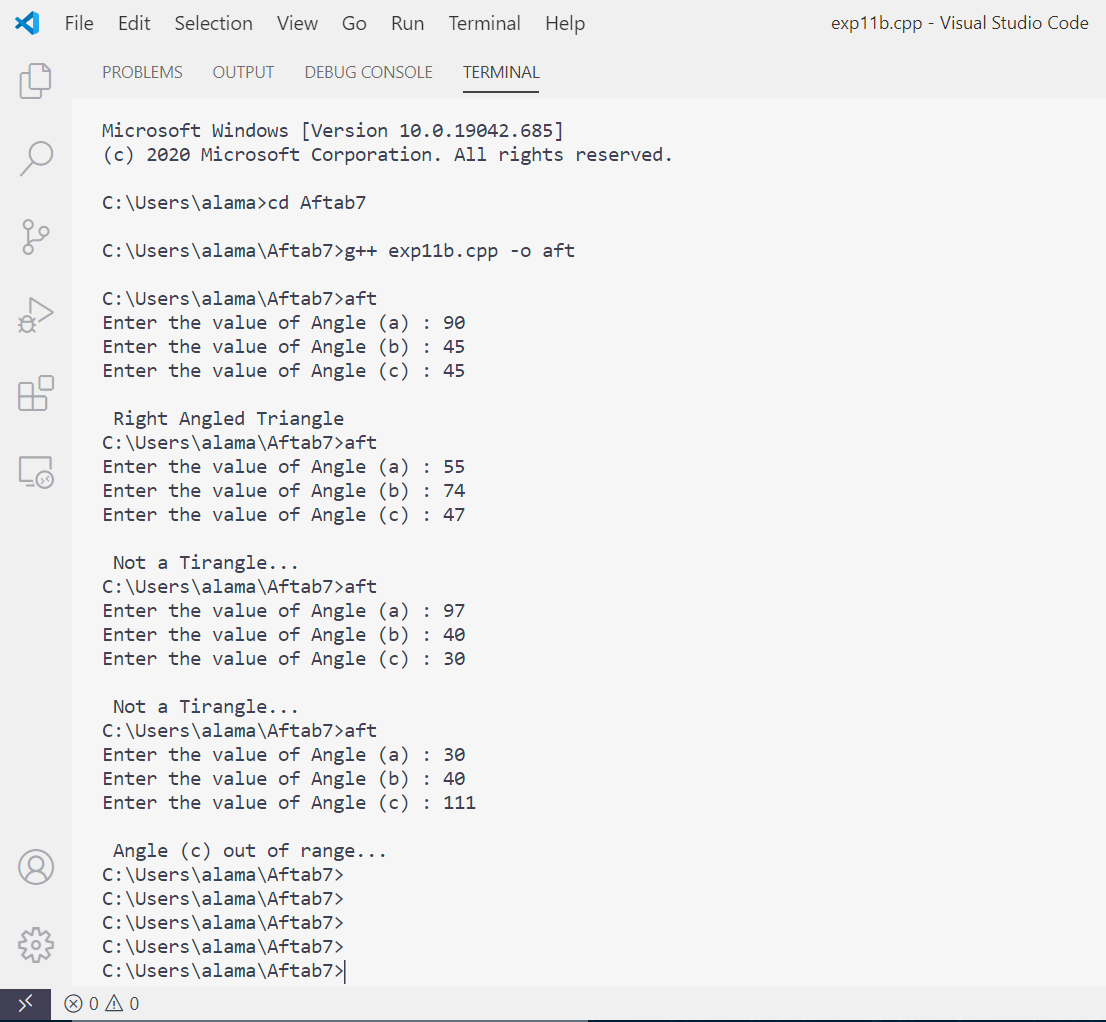
**40.**        cout << "\n Angle (a) out of range...";

**41.**    }

**42.**    return 0;

**43.**}

**Output Screenshot :**



**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}**