National University of Science and Technology

School of Mechanical and Manufacturing Engineering

HU-117 Fundamental of programming

English Assignment:
Teacher: Affan

Introduction:

Name	CMS
Muhammad Uzair	466092

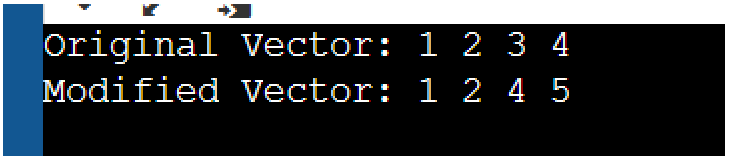
Section: ME-15B

Date: 26-12-2023

Task No: 1

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

int main() {

    vector<int> myVector = {1, 2, 3, 4};
    cout << "Original Vector: ";
    for (auto it = myVector.begin(); it != myVector.end(); ++it) {
        cout << *it << " ";
    }
    cout << std::endl;
    myVector.push_back(5);
    if (!myVector.empty()) {
            int positionToRemove = 2;
        myVector.erase(myVector.begin() + positionToRemove);
    }


    cout << "Modified Vector: ";
    for (const auto& element : myVector) {
        cout << element << " ";
    }
    cout <<endl;

    return 0;
}
```

Output:



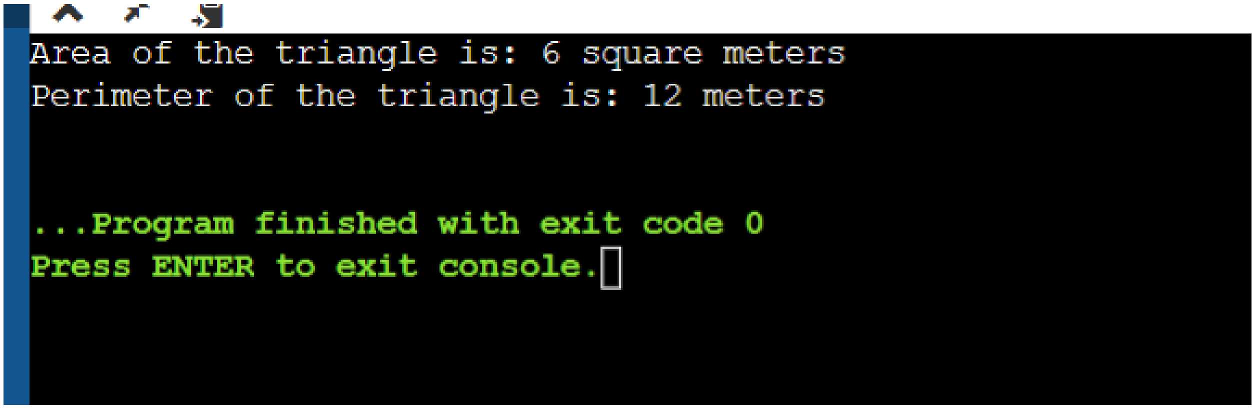
```
Original Vector: 1 2 3 4
Modified Vector: 1 2 4 5
```

Task No:2

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

// Define a class named Triangle
class Triangle {
  // Declare the private attributes to store the side lengths
  private:
    float a, b, c;

  // Declare the public methods to access and modify the attributes
  public:
    // A constructor to initialize the side lengths
    Triangle(float x, float y, float z) {
      a = x;
      b = y;
      c = z;
    }

    // A method to print the area of the triangle
    void print_area() {
      // Use Heron's formula to calculate the area
      float s = (a + b + c) / 2;
      float area = sqrt(s * (s - a) * (s - b) * (s - c));
      cout << "Area of the triangle is: " << area << " square meters" << endl;
    }

    // A method to print the perimeter of the triangle
    void print_perimeter() {
      // Add the side lengths to get the perimeter
      float perimeter = a + b + c;
      cout << "Perimeter of the triangle is: " << perimeter << " meters" << endl;
    }
};

// The main function
int main() {
  // Create an object of the Triangle class with the given side lengths
  Triangle t(3, 4, 5);

  // Call the methods to print the area and perimeter
  t.print_area();
  t.print_perimeter();

  return 0;
```

```
}
```

Output:

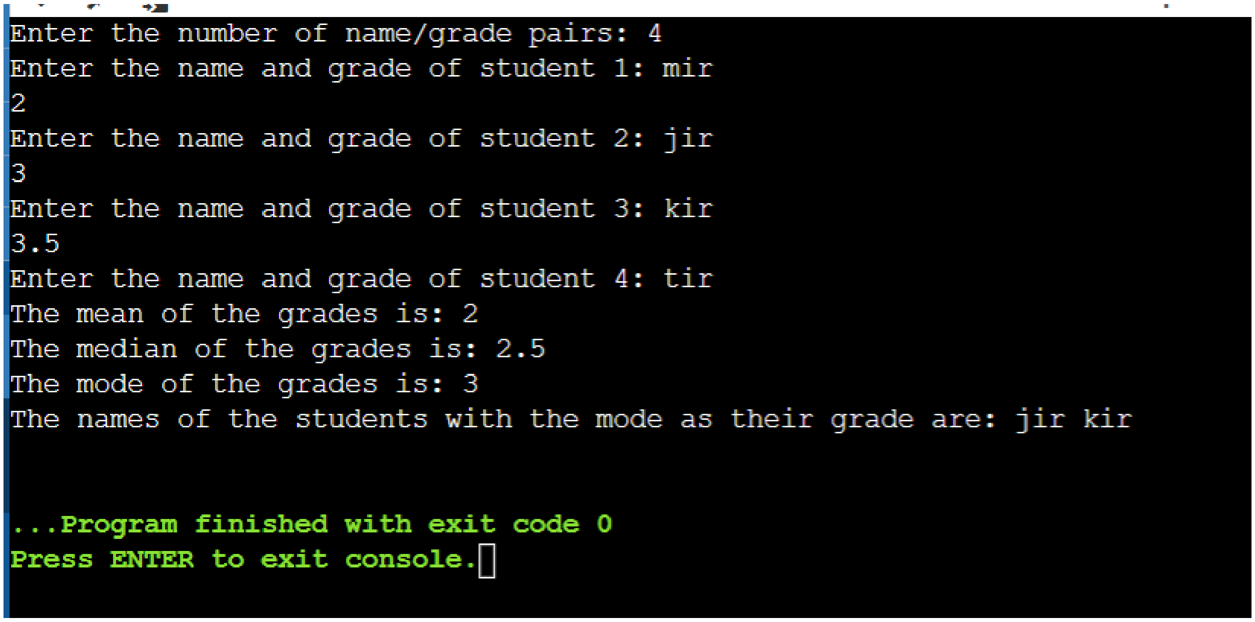


```
Area of the triangle is: 6 square meters
Perimeter of the triangle is: 12 meters


...Program finished with exit code 0
Press ENTER to exit console.
```

Task no:3

```
#include <iostream>

#include <vector>

#include <algorithm>

#include <map>

using namespace std;


// A function to calculate the mean of a vector of integers

double mean(const vector<int>& v) {

   int sum = 0;

   for (int x : v) {

      sum += x;

   }

   return (double)sum / v.size();

}
```

```
// A function to calculate the median of a vector of integers
double median(vector<int> v) {
    sort(v.begin(), v.end()); // Sort the vector in ascending order
    int n = v.size();
    if (n % 2 == 0) { // If the size is even, the median is the average of the middle two elements
        return (v[n/2] + v[n/2 - 1]) / 2.0;
    }
    else { // If the size is odd, the median is the middle element
        return v[n/2];
    }
}

// A function to calculate the mode of a vector of integers
int mode(const vector<int>& v) {
    map<int, int> freq; // A map to store the frequency of each element
    int max_freq = 0; // The maximum frequency so far
    int mode = 0; // The mode so far
    for (int x : v) {
        freq[x]++; // Increment the frequency of x
        if (freq[x] > max_freq) { // If the frequency of x is greater than the maximum frequency, update the mode and the maximum frequency
            max_freq = freq[x];
            mode = x;
        }
    }
    return mode;
}

int main() {
    int n; // The number of name/grade pairs
    cout << "Enter the number of name/grade pairs: ";
    cin >> n;

    vector<string> names(n); // A vector to store the names
    vector<int> grades(n); // A vector to store the grades

    // Loop to input the name/grade pairs
    for (int i = 0; i < n; i++) {
        cout << "Enter the name and grade of student " << i + 1 << ": ";
        cin >> names[i] >> grades[i];
    }

    // Display the mean, median and mode of the grades
    cout << "The mean of the grades is: " << mean(grades) << endl;
    cout << "The median of the grades is: " << median(grades) << endl;
    cout << "The mode of the grades is: " << mode(grades) << endl;

    // Display the names of the students with the mode as their grade
    cout << "The names of the students with the mode as their grade are: ";
    for (int i = 0; i < n; i++) {
        if (grades[i] == mode(grades)) { // If the grade of the i-th student is equal to the mode, print their name
            cout << names[i] << " ";
        }
    }
    cout << endl;


    return 0;
}
```



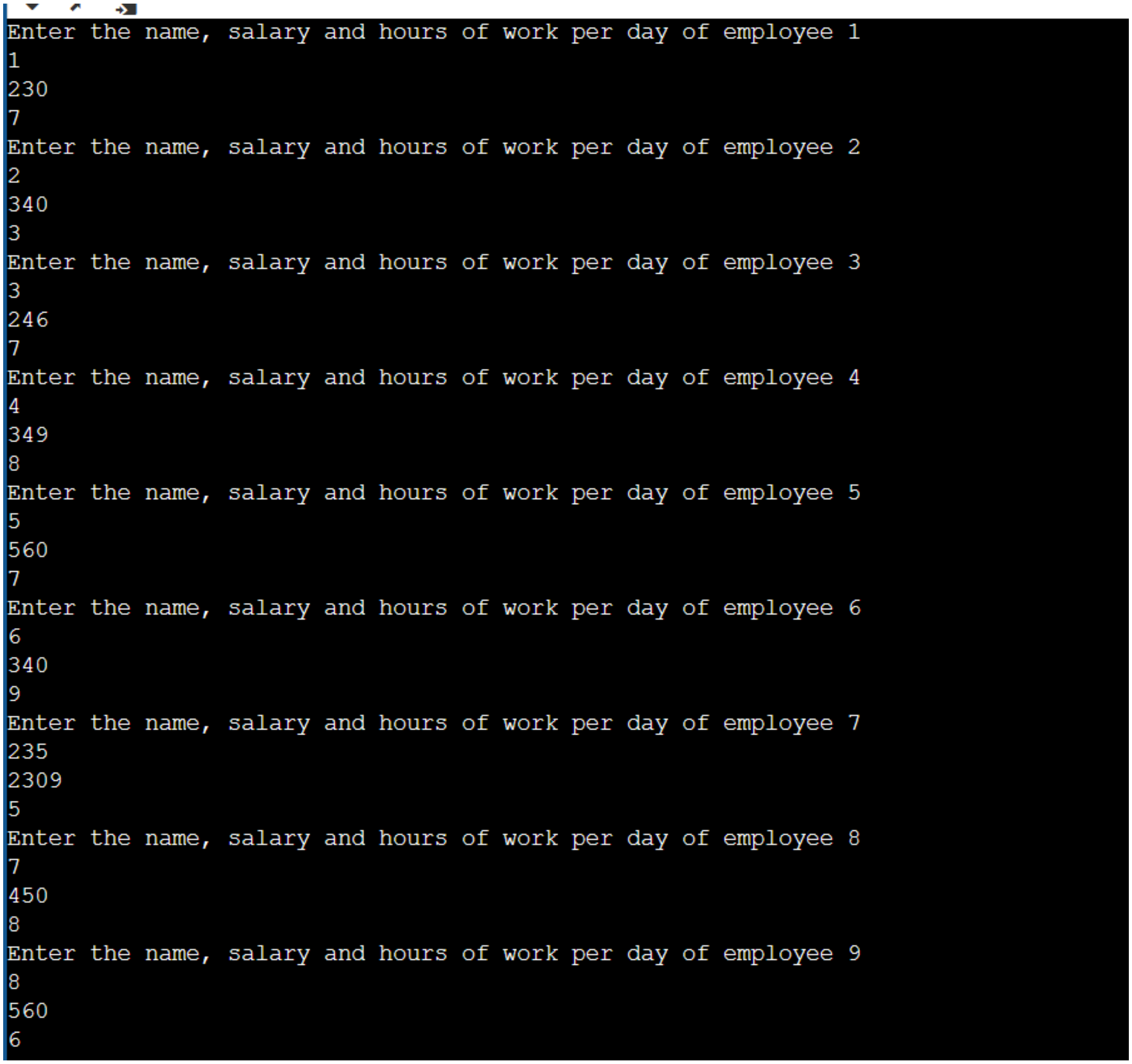
```
Enter the number of name/grade pairs: 4
Enter the name and grade of student 1: mir
2
Enter the name and grade of student 2: jir
3
Enter the name and grade of student 3: kir
3.5
Enter the name and grade of student 4: tir
The mean of the grades is: 2
The median of the grades is: 2.5
The mode of the grades is: 3
The names of the students with the mode as their grade are: jir kir


...Program finished with exit code 0
Press ENTER to exit console.
```

Task No: 4

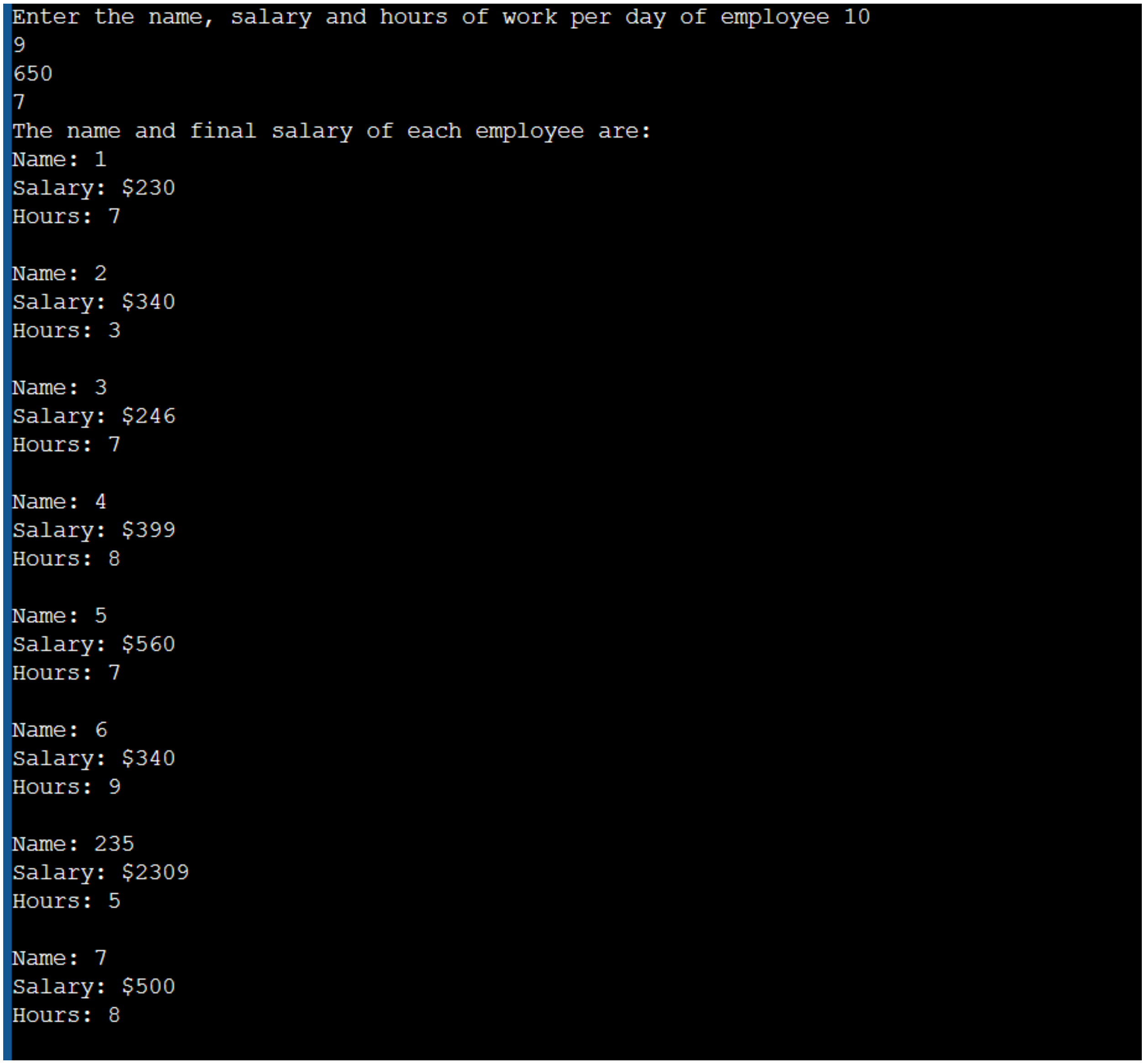
```
#include <iostream>
```

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

// Define a structure to store employee information
struct Employee {
  string name; // Use string instead of char to store names
  int salary;
  int hours;
};

// Define a function to increase the salary based on hours
void increase_salary(Employee& e) {
  if (e.hours == 8) {
    e.salary += 50; // Increase salary by $50
  }
  else if (e.hours == 10) {
    e.salary += 100; // Increase salary by $100
  }
  else if (e.hours >= 12) {
    e.salary += 150; // Increase salary by $150
  }
}

// Define a function to print employee information
```

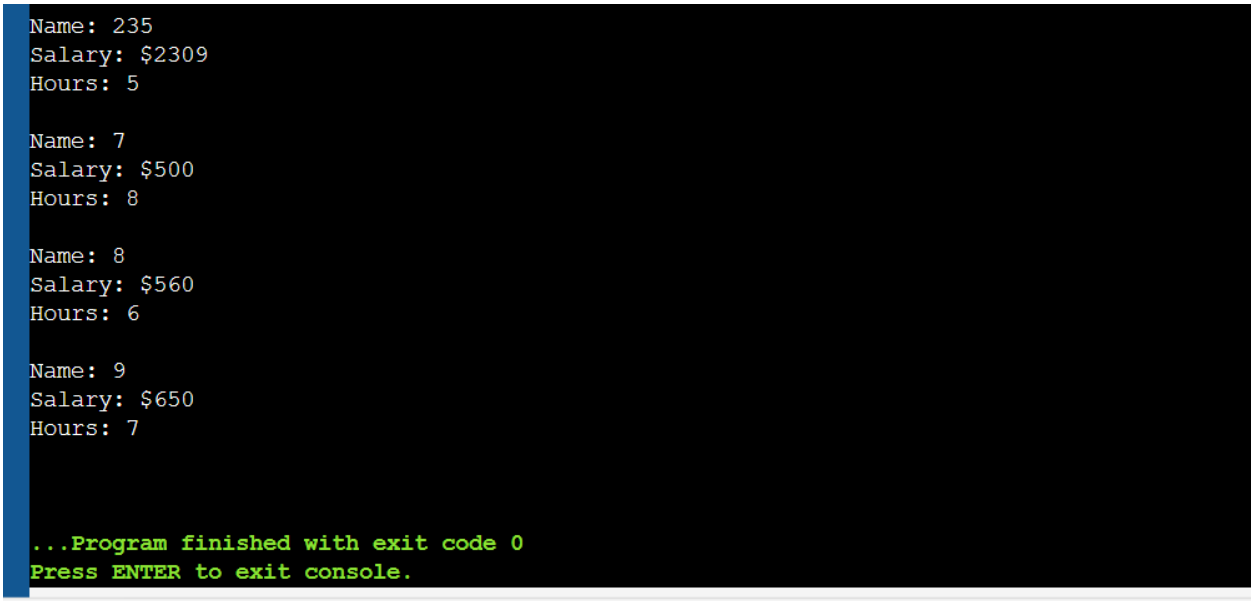
```
void print_employee(const Employee& e) {
  cout << "Name: " << e.name << endl;
  cout << "Salary: $" << e.salary << endl;
  cout << "Hours: " << e.hours << endl;
  cout << endl;
}

int main() {
  // Declare an array of 10 employees
  Employee employees[10];

  // Input employee information from the user
  for (int i = 0; i < 10; i++) {
    cout << "Enter the name, salary and hours of work per day of employee " << i + 1 << endl;
    cin >> employees[i].name >> employees[i].salary >> employees[i].hours;
  }

  // Increase the salary of each employee
  for (int i = 0; i < 10; i++) {
    increase_salary(employees[i]);
  }

  // Print the name and final salary of each employee
  cout << "The name and final salary of each employee are:" << endl;
  for (int i = 0; i < 10; i++) {
    print_employee(employees[i]);
  }

  return 0;
}
```



```
Enter the name, salary and hours of work per day of employee 1
1
230
7
Enter the name, salary and hours of work per day of employee 2
2
340
3
Enter the name, salary and hours of work per day of employee 3
3
246
7
Enter the name, salary and hours of work per day of employee 4
4
349
8
Enter the name, salary and hours of work per day of employee 5
5
560
7
Enter the name, salary and hours of work per day of employee 6
6
340
9
Enter the name, salary and hours of work per day of employee 7
235
2309
5
Enter the name, salary and hours of work per day of employee 8
7
450
8
Enter the name, salary and hours of work per day of employee 9
8
560
6
```



```
Enter the name, salary and hours of work per day of employee 10
9
650
7
The name and final salary of each employee are:
Name: 1
Salary: $230
Hours: 7

Name: 2
Salary: $340
Hours: 3

Name: 3
Salary: $246
Hours: 7

Name: 4
Salary: $399
Hours: 8

Name: 5
Salary: $560
Hours: 7

Name: 6
Salary: $340
Hours: 9

Name: 235
Salary: $2309
Hours: 5

Name: 7
Salary: $500
Hours: 8
```



```
Name: 235
Salary: $2309
Hours: 5

Name: 7
Salary: $500
Hours: 8

Name: 8
Salary: $560
Hours: 6

Name: 9
Salary: $650
Hours: 7



...Program finished with exit code 0
Press ENTER to exit console.
```