

# **Fundamentals of Programming**

## **Lab manual 10**

**ME-15**

**Section A**

**Faizan Ahmad**

**(476602)**

Task 1:

1. Iterate Through Vector Using Iterators and print all pushed elements. Next you need to push integer 5 and remove element at that position.

Code:

```
#include<iostream>
```

```
#include<vector>
```

```
using namespace std;
```

```
int main() {
```

```
vector<int> a;
```

```
for (int i = 1; i<=10; i++ ) {
```

```
    a.push_back(i);
```

```
}
```

```
cout<<"The elements of the vector 'a' are: ";
```

```
for (auto i = a.begin(); i != a.end(); i++ ) {
```

```
    cout<< *i <<" ";
```

```
}
```

```
cout<<endl;
```

```
a.push_back(5);
```

```
// Print elements after pushing 5
cout << "\nElements after pushing 5: ";
for (auto it = a.begin(); it != a.end(); ++it){

    cout << *it << " ";
}
cout<<endl;

a.erase(a.begin()+4);

cout<<"elements after erasing the 5th element: ";
for (auto i = a.begin(); i != a.end(); i++){
    cout<< *i << " ";
}

return 0;
}
```

Execute:

```
The elements of the vector 'a' are: 1 2 3 4 5 6 7 8 9 10
Elements after pushing 5: 1 2 3 4 5 6 7 8 9 10 5
elements after erasing the 5th element: 1 2 3 4 6 7 8 9 10 5
-----
Process exited after 0.054 seconds with return value 0
Press any key to continue . . .
```

Task 2:

1. Write a complete C++ program that uses 2 vectors, 1 for names (string) and 1 for grades (int)
  - a. Ask the user for the number of name/grade pairs that will be entered.
  - b. Display the mean of the grades.
  - c. Display the median of the grades.
  - d. Display the mode of the grades.
  - e. Display the names of the students with the mode as their grade.

Code:

```
#include <iostream>
#include <vector>
#include <string>
#include <algorithm>
```

```
using namespace std;
```

```
int main() {
    vector<string> names;
    vector<int> grades;
    int size;
```

```
    cout << "Enter the number of students: ";
```

```
cin >> size;
```

```
// Loop to input names and grades
```

```
for (int i = 1; i <= size; ++i) {  
    cout << "Enter the name of student " << i << " : ";  
    string name;  
    cin >> name;  
    names.push_back(name);  
    cout << "Enter the grade of student " << i << " : ";  
    int marks;  
    cin >> marks;  
    grades.push_back(marks);  
}
```

```
float sum = 0;
```

```
for (auto i = grades.begin(); i != grades.end(); ++i) {  
    sum += *i;  
}
```

```
float mean = sum / size;
```

```
cout << endl << "The mean grade of the students is: " << mean << endl;
```

```
// Calculate median
```

```
vector<int> temp_grade = grades;  
sort(temp_grade.begin(), temp_grade.end());
```

```
float median;
```

```

size_t mid = size / 2;
if (size % 2 != 0) {
    median = temp_grade[mid];
} else {
    median = static_cast<float>(temp_grade[mid - 1] + temp_grade[mid]) / 2.0;
}

cout << "The median grade of the students is: " << median << endl;

// Calculate mode
int mode = temp_grade[0];
int maxFreq = 1;
int currentNum = temp_grade[0];
int currentFreq = 1;

for (int i = 1; i < temp_grade.size(); ++i) {
    if (temp_grade[i] == currentNum) {
        ++currentFreq;
    } else {
        currentNum = temp_grade[i];
        currentFreq = 1;
    }

    if (currentFreq > maxFreq) {
        maxFreq = currentFreq;
        mode = currentNum;
    }
}

```

```
cout << "Mode grade: " << mode << " (appeared " << maxFreq << " times)" << endl;
```

```
cout << endl << "Entered names and grades:" << endl;
```

```
for (size_t i = 0; i < names.size(); ++i) {
```

```
    cout << names[i] << ": " << grades[i] << endl;
```

```
}
```

```
cout << endl << "The students with the grade equal to the mode are:" << endl;
```

```
for (size_t i = 0; i < size; ++i) {
```

```
    if (grades[i] == mode) {
```

```
        cout << names[i] << endl;
```

```
    }
```

```
}
```

```
return 0;
```

```
}
```

Execute:

```
Enter the number of students: 5
Enter the name of student 1 : Faizan
Enter the grade of student 1 : 50
Enter the name of student 2 : Ahmad
Enter the grade of student 2 : 45
Enter the name of student 3 : Taha
Enter the grade of student 3 : 45
Enter the name of student 4 : Abdullah
Enter the grade of student 4 : 40
Enter the name of student 5 : Hashim
Enter the grade of student 5 : 55

The mean grade of the students is: 47
The median grade of the students is: 45
Mode grade: 45 (appeared 2 times)

Entered names and grades:
Faizan: 50
Ahmad: 45
Taha: 45
Abdullah: 40
Hashim: 55

The students with the grade equal to the mode are:
Ahmad
Taha
```

### Task 3

Write a program to print the area and perimeter of a triangle having sides of 3 m, 4 m and 5 m by creating a class named 'Triangle' with a function to print the area and perimeter.

Code

```
#include <iostream>
```

```
#include <cmath>
```

```
using namespace std;
```

```
class Triangle {
```

```
private:
```

```
    double side1;
```

```
    double side2;
```

```
    double side3;
```



public:

```
Triangle(double s1, double s2, double s3) : side1(s1), side2(s2), side3(s3) {}
```

```
double perimeter() {  
    return side1 + side2 + side3;  
}
```

```
double area() {  
    // Heron's formula for the area of a triangle  
    double s = perimeter() / 2;  
    return sqrt(s * (s - side1) * (s - side2) * (s - side3));  
}  
};
```

```
int main() {  
    // Create an instance of the Triangle class  
    Triangle triangle(3, 4, 5);  
  
    // Calculate and print the area and perimeter  
    cout << "Perimeter: " << triangle.perimeter() << " m" << endl;  
    cout << "Area: " << triangle.area() << " m²" << endl;  
  
    return 0;  
}
```

Execute:

```
Perimeter: 12 m
Area: 6 m²

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

Task 4:

1. Write a structure to store the names, salary, and hours of work per day of 10 employees in a company. Write a program to increase the salary depending on the number of hours of work per day as follows and then print the name of all the employees along with their final salaries.

Code:

```
#include <iostream>
```

```
#include <iomanip>
```

```
#include <string>
```

```
using namespace std;
```

```
// Define the structure to store employee information
```

```
struct Employee {
```

```
    string name;
```

```
    double salary;
```

```
    int hoursWorked;
```

```
};
```

```
// Function to calculate and update salary based on hours worked
```

```
void updateSalary(Employee& emp) {
```

```
    // Increase salary based on hours worked
```

```
    if (emp.hoursWorked > 40) {  
        emp.salary += (emp.hoursWorked - 40) * 10; // Assuming an overtime rate of $10 per extra hour  
    }  
}
```

```
int main() {  
    const int numEmployees = 10;  
  
    // Create an array of Employee structures  
    Employee employees[numEmployees];  
  
    // Input employee information  
    for (int i = 0; i < numEmployees; ++i) {  
        cout << "Enter details for Employee " << i + 1 << " : " << endl;  
        cout << "Name: ";  
        getline(cin, employees[i].name);  
  
        cout << "Salary: $";  
        cin >> employees[i].salary;  
  
        cout << "Hours worked per day: ";  
        cin >> employees[i].hoursWorked;  
  
        // Consume the newline character left in the buffer  
        cin.ignore();  
        cout << endl;  
  
        // Update salary based on hours worked  
        updateSalary(employees[i]);  
    }  
}
```

```

}

// Output employee information
cout << "Employee Information:" << endl;
cout << setw(20) << left << "Name" << setw(15) << "Salary" << endl;
cout << setfill('-') << setw(35) << "" << setfill(' ') << endl;

for (int i = 0; i < numEmployees; ++i) {
    cout << setw(20) << left << employees[i].name << setw(15) << fixed << setprecision(2) <<
employees[i].salary << endl;
}

return 0;
}

```

Execute:

Enter details for Employee 1:

Name: Faizan

Salary: \$400

Hours worked per day: 4

Enter details for Employee 2:

Name: Hulu

Salary: \$500

Hours worked per day: 5

Enter details for Employee 3:

Name: E.

Salary: \$600

Hours worked per day: 1

Enter details for Employee 4:

Name: hmm

Salary: \$500

Hours worked per day: 7

Enter details for Employee 5:

Name: nofil

Salary: \$50000

Hours worked per day: 1

Enter details for Employee 6:

Name: dfshlj

Salary: \$4

Hours worked per day: 4

Enter details for Employee 7:

```
Salary: $4
Hours worked per day: 5

Enter details for Employee 8:
Name: jihgre
Salary: $3
Hours worked per day: 4

Enter details for Employee 9:
Name: jerh
Salary: $5
Hours worked per day: 6

Enter details for Employee 10:
Name: peort
Salary: $4
Hours worked per day: 6
```

Employee Information:

Name	Salary
Faizan	400.00
Hulu	500.00
E.	600.00
hmm	500.00
nofil	50000.00
dfshlj	4.00
eijgr	4.00
jihgre	3.00
jerh	5.00
peort	4.00