CSS-114- FUNDAMENTALS OF PROGRAMMING

LAB MANUAL #10

LAB AND HOME TASK

Course Instructor: Dr. Jawad Khan

Lab Instructor: Muhammad Affan

Student Name: ABDULLAH BIN KHORRAM

CMS ID: 466612

DATE: 23 DECEMBER 2023.

Tasks (Due in last week of December):

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:

```cpp
#include <iostream>
#include <vector>

using namespace std;

int main()
{
    vector<int> V1;

    for (int i = 1; i <= 10; i++)
       { V1.push_back(i); }

    cout << "Output of vector using begin and end: ";
    for (auto i = V1.begin(); i != V1.end(); i++)
       {
        cout << *i << " "; }
        cout<<endl;

    cout << "Output of vector using cbegin and cend: ";
    for (auto i = V1.cbegin(); i != V1.cend(); ++i)
       {
        cout << *i << " "; }
        cout<<endl;

    cout << "Output of vector rbegin and rend: ";
    for (auto i = V1.rbegin(); i != V1.rend(); ++i)
     {
       cout << *i << " "; }
         cout<<endl;

    cout << "Output of vector crbegin and crend : ";
    for (auto i = V1.crbegin(); i != V1.crend(); ++i)
      {  cout << *i << " "; }
 cout<<endl;

     V1.pop_back();
     V1.push_back(5);
    cout << "The New Vector elements are: ";
    for (int i = 0; i < V1.size(); i++)
      {  cout << V1[i] << " "; }



    return 0;
}
```

OUTPUT:

```
Output

/tmp/iYHoNgVlTt.o
Output of vector using begin and end: 1 2 3 4 5 6 7 8 9 10
Output of vector using cbegin and cend: 1 2 3 4 5 6 7 8 9 10
Output of vector rbegin and rend: 10 9 8 7 6 5 4 3 2 1
Output of vector crbegin and crend : 10 9 8 7 6 5 4 3 2 1
The New Vector elements are: 1 2 3 4 5 6 7 8 9 5
```

2. 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.

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

using namespace std;

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

    // Input names and grades into vectors
    vector<string> names(pairs);
    vector<float> grades(pairs);
    float sum = 0;
    cout << "Enter the name/grade pairs~" << endl;
    cout<<endl;
    for (size_t i = 0; i < pairs; ++i) {
        cout << "Enter the name for student " << i + 1 << ": ";
        cin >> names[i];
        cout << "Enter the grade of student " << i + 1 << ": ";
        cin >> grades[i];
        cout<<endl;
    }
```

```cpp
//finding mean of the grades
    for(int i=0; i<pairs; i++){
       sum = grades[i] + sum;
    }
    float mean = sum / pairs;
    cout<<endl;
     cout<<"Mean of the grades is:"<<mean<<endl;
    //if grades are equal to median, print names of students
    cout<<"Students, if any, whose grades are equal to the mean: ";
     for(int i=0;i<pairs;i++){
         if(grades[i]==mean){
             cout<<names[i]<<" ";
         }
     } cout<<endl;
//finding median of the grades
//first we sort the vector
    for (size_t i = 0; i < pairs - 1; ++i) {
        for (size_t j = 0; j < pairs - i - 1; ++j) {
            if (grades[j] > grades[j + 1]) {
                // Swap elements if they are in the wrong order
                swap(grades[j], grades[j + 1]);
            }
        }
```

```
48      }   cout<<"The Sorted vector is: "<<endl;
49      for(int i=0;i<pairs;i++){
50          cout<<grades[i]<<" "; }
51
52      float median,halfpairs = (pairs / 2) -1; //halfpairs refers to the half
            position of the vector
53      cout<<endl;
54
55     //there are two ways to find the median, if the number of grades is even or odd
56     //if even, middle two values are added and sum is halved
57      if(pairs%2==0){
58          median = (grades[halfpairs] + grades[halfpairs+1]) /2;
59      }
60  //if odd, middle value is median
61     if(pairs%2!=0){
62        median = grades[halfpairs+1.5];
63     }
64     cout<<"The Median is:"<<median<<endl;
```

```
65      //finding the mode
66      int freq;
67      //print all repeated elements
68      for(int i=0;i<pairs-1; i++){
69          if(grades[i]==grades[i+1]){ freq++;
70              cout<<grades[i]<<" "; }
71      } cout<<endl;
72      if(freq!=0){cout<<"The value(s) which frequents the most times in the above
            values is the mode"<<endl;}
73      if(freq==0){ cout<<"There is no Mode"<<endl;}
74
75      return 0;
76  }
```

OUTPUT:

Output

```
/tmp/43YgjlnH3w.o
Enter the number of name/grade pairs: 3
Enter the name/grade pairs~

Enter the name for student 1: Ashley
Enter the grade of student 1: 30
Enter the name for student 2: Sam
Enter the grade of student 2: 40
Enter the name for student 3: Josh
Enter the grade of student 3: 20
Mean of the grades is:30
Students, if any, whose grades are equal to the mean: Ashley
The Sorted vector is:
20 30 40
The Median is:30

There is no Mode
```

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:

```cpp
#include <iostream>
#include <cmath>

using namespace std;

class Triangle {
private:
    double side1, side2, side3;

public:
    // Constructor to initialize the sides of the triangle
    Triangle(double s1, double s2, double s3) : side1(s1), side2(s2), side3(s3)
        {}

    // Function to calculate the perimeter of the triangle
    double calculatePerimeter() const {
        return side1 + side2 + side3;
    }

    // Function to calculate the area of the triangle using the known formula
    double calculateArea() const {
        double s = calculatePerimeter() / 2.0;
        return sqrt(s * (s - side1) * (s - side2) * (s - side3));
    }

    // Function to print the area and perimeter of the triangle
    void printDetails() const {
        cout << "Triangle Details:" << endl;
        cout << "Side 1: " << side1 << " meters" << endl;
        cout << "Side 2: " << side2 << " meters" << endl;
        cout << "Side 3: " << side3 << " meters" << endl;
        cout << "Perimeter: " << calculatePerimeter() << " m" << endl;
        cout << "Area: " << calculateArea() << " square meters" << endl;
    }
};

int main() {
    // Create a Triangle object with sides 3 m, 4 m, and 5 m
    Triangle myTriangle(3.0, 4.0, 5.0);

    // Print the details (area and perimeter) of the triangle
    myTriangle.printDetails();

    return 0;
}
```

OUTPUT:

Output

```
/tmp/43YgjlnH3w.o
Triangle Details:
Side 1: 3 meters
Side 2: 4 meters
Side 3: 5 meters
Perimeter: 12 m
Area: 6 square meters
```

4. 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.

Hours of work per day	8	10	>=12
Increase in Salary	$50	$100	$150

CODE:

```cpp
#include <iostream>
#include <string>

using namespace std;

// Structure to store employee information
struct Employee {
    string name;
    double salary;
    int hoursWorkedPerDay;
};

// Function to increase salary based on hours worked per day
void increaseSalary(Employee& emp) {
    if (emp.hoursWorkedPerDay >= 12) {
        emp.salary += 150.0;
    } else if (emp.hoursWorkedPerDay >= 10) {
        emp.salary += 100.0;
    } else if (emp.hoursWorkedPerDay >= 8) {
        emp.salary += 50.0;
    }
}

int main() {
    int numEmployees = 10;

    // Create an array of Employee structures
    Employee employees[numEmployees];

    // Input salary and hours worked per day for each employee
    for (int i = 0; i < numEmployees; ++i) {
        cout << "Enter name for Employee " << i + 1 << ": ";
        cin>>employees[i].name;

        cout << "Enter salary for Employee " << i + 1 << ": $";
        cin >>employees[i].salary;

        cout << "Enter hours worked per day for Employee " << i + 1 << ": ";
        cin >>employees[i].hoursWorkedPerDay;

        cin.ignore(); // Clear the newline character from the buffer
        cout << endl;
    }

    // Increase salary for each employee based on hours worked
    for (int i = 0; i < numEmployees; ++i) {
        increaseSalary(employees[i]);
    }

```

```
50      // Print the names and final salaries of all employees
51      cout << "Name            Final Salary" << endl;
52      cout << "----------------------------" << endl;
53
54      for (int i = 0; i < numEmployees; ++i) {
55          cout << employees[i].name << "            $" << employees[i].salary <<
                endl;
56      }
57
58      return 0;
59  }
```

OUTPUT:

Output

```
/tmp/9zhQOhgxtB.o
Enter name for Employee 1: Ashley
Enter salary for Employee 1: $500
Enter hours worked per day for Employee 1: 12
Enter name for Employee 2: J
onathanEnter salary for Employee 2: $800
Enter hours worked per day for Employee 2: 8
Enter name for Employee 3: Marcus
Enter salary for Employee 3: $560
Enter hours worked per day for Employee 3: 10
Enter name for Employee 4: Sarah
Enter salary for Employee 4: $1750
Enter hours worked per day for Employee 4: 7
Enter name for Employee 5: Anthony
Enter salary for Employee 5: $800
Enter hours worked per day for Employee 5: 12
Enter name for Employee 6: Alisha
Enter salary for Employee 6: $1800
Enter hours worked per day for Employee 6: 13
Enter name for Employee 7: Sam
Enter salary for Employee 7: $650
Enter hours worked per day for Employee 7: 9
Enter name for Employee 8: Zain
Enter salary for Employee 8: $780
Enter hours worked per day for Employee 8: 11
Enter name for Employee 9: zack
Enter salary for Employee 9: $900
Enter hours worked per day for Employee 9: 10
Enter name for Employee 10: Sebastian
Enter salary for Employee 10: $2100
Enter hours worked per day for Employee 10: 8
Name            Final Salary
----------------------------
Ashley            $650
Jonathan            $850
Marcus            $660
Sarah            $1750
Anthony            $950
Alisha            $1950
Sam            $700
Zain            $880
zack            $1000
Sebastian            $2150
```