

# CS & Programming Lab

## Lab Manual 10

---

Name: Abdullah Jamil Tung

Class: ME-15-Sec-A

Roll #: 478459

## **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.\*/

```
#include<iostream>
```

```
#include<vector>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    //Declaring variables
```

```
    vector<int> v;
```

```
    //Taking inputs
```

```
    for (int i=0; i<10; i++)
```

```
    {
```

```
        v.push_back(i);
```

```
    }
```

```
    //Iterating and printing the elements
```

```
    cout<<"The elements of vector 'v' = ";
```

```
for (auto i=v.begin(); i != v.end(); ++i)
{
    cout<<*i<<" ";
}
cout<<endl;

//Pushing "5" at the end and removing the element
v.push_back(5);
v.erase(v.end()-2);

//Displaying the vector after editing
cout<<"The elements of vector 'v' after editing = ";
for (auto i=v.begin(); i != v.end(); ++i)
{
    cout<<*i<<" ";
}
}
```

```

1  /*1. Iterate Through Vector Using Iterators and print all pushed elements.
2     Next you need to push integer 5 and remove element at that position.*/
3
4  #include<iostream>
5  #include<vector>
6  using namespace std;
7
8  int main()
9  {
10     //Declaring variables
11     vector<int> v;
12
13     //Taking inputs
14     for (int i=0; i<10; i++)
15     {
16         v.push_back(i);
17     }
18
19     //Iterating and printing the elements
20     cout<<"The elements of vector 'v' = ";
21     for (auto i=v.begin(); i != v.end(); ++i)
22     {
23         cout<<*i<<" ";
24     }
25     cout<<endl;
26
27     //Pushing "5" at the end and removing the element
28     v.push_back(5);
29     v.erase(v.end()-2);
30
31     //Displaying the vector after editing
32     cout<<"The elements of vector 'v' after editing = ";
33     for (auto i=v.begin(); i != v.end(); ++i)
34     {
35         cout<<*i<<" ";
36     }
37 }

```

/tmp/hV0Mejap1L.o

The elements of vector 'v' = 0 1 2 3 4 5 6 7 8 9

The elements of vector 'v' after editing = 0 1 2 3 4 5 6 7 8 5

## **Task #2**

/\*2. Write a complete C++ program that uses 2 vectors, 1 for names (string) and 1 for grades (int)\*/

```
#include <iostream>
```

```
#include <vector>
```

```
#include <algorithm>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    //a. Ask the user for the number of name/grade pairs that will be entered.
```

```
    int n;
```

```
    cout<<"Enter the number of name/grade pairs: ";
```

```
    cin>>n;
```

```
    //Declaring variables
```

```
    vector<string> names;
```

```
    vector<int> grades;
```

```
    //Taking inputs
```

```
for (int i = 0; i<n; i++)
{
    cout<<endl;
    string name;
    int grade;

    cout<<"Enter name #" <<i + 1<< ": ";
    cin>>name;

    cout<<"Enter grade #" <<i + 1<< ": ";
    cin>>grade;

    names.push_back(name);
    grades.push_back(grade);
}
cout<<endl;

//b. Display the mean of the grades.
float mean = 0.0;
float sum;

for (int i=0; i<grades.size(); i++)
{
    sum+=grades[i];
```

```
}
```

```
mean= float (sum/grades.size());
```

```
cout<<"Mean = "<<mean<<endl;
```

```
//c. Display the median of the grades.
```

```
float median = 0.0;
```

```
if (!grades.empty())
```

```
{sort(grades.begin(), grades.end());
```

```
size_t size = grades.size();
```

```
if (size%2 == 0)
```

```
{median = (grades[size / 2 - 1] + grades[size / 2]) / 2.0;}
```

```
else
```

```
{median = grades[size / 2];}
```

```
}
```

```
cout<<"Median = "<<median<<endl;
```

```
//d. Display the mode of the grades
```

```
sort(grades.begin(), grades.end());;
```

```
int mode=0, max_count=0, count=1;
```

```
for (size_t i = 1; i < grades.size(); ++i)
{
    if (grades[i] == grades[i - 1])
    { ++count; }
    else
    { count = 1; }
```

```
if (count > max_count)
    { max_count = count;
mode = grades[i]; }
}
```

```
cout << "Mode = " << mode << endl;
```

//e. Display the names of the students with the mode as their grade.

```
cout << "\nStudents with the mode as their grade: ";
```

```
for (size_t i = 0; i < grades.size(); ++i)
{ if (grades[i] == mode)
    { cout << names[grades.size()-(i+1)] << " "; }
}
```

```
cout << endl;
```

```
return 0; }
```



```

1  /*2. Write a complete C++ program that uses 2 vectors, 1 for names (string) and 1 for grades (int)*/
2
3  #include <iostream>
4  #include <vector>
5  #include <algorithm>
6
7  using namespace std;
8
9  int main()
10 {
11     //a. Ask the user for the number of name/grade pairs that will be entered.
12     int n;
13     cout<<"Enter the number of name/grade pairs: ";
14     cin>>n;
15
16     //Declaring variables
17     vector<string> names;
18     vector<int> grades;
19
20     //Taking inputs
21     for (int i = 0; i<n; i++)
22     {
23         cout<<endl;
24         string name;
25         int grade;
26
27         cout<<"Enter name #"<<i + 1<<" ": ";
28         cin>>name;
29
30         cout<<"Enter grade #" <<i + 1<<": ";
31         cin>>grade;
32
33         names.push_back(name);
34         grades.push_back(grade);
35     }
36     cout<<endl;
37

```

```

38     //b. Display the mean of the grades.
39     float mean = 0.0;
40     float sum;
41     for (int i=0; i<grades.size(); i++)
42     {
43         sum+=grades[i];
44     }
45
46     mean= float (sum/grades.size());
47     cout<<"Mean = "<<mean<<endl;
48
49     //c. Display the median of the grades.
50     float median = 0.0;
51
52     if (!grades.empty())
53     {sort(grades.begin(), grades.end());
54       size_t size = grades.size();
55
56       if (size%2 == 0)
57       {median = (grades[size / 2 - 1] + grades[size / 2]) / 2.0;}
58
59       else
60       {median = grades[size / 2];}
61     }
62
63     cout<<"Median = "<<median<<endl;
64
65     //d. Display the mode of the grades
66     sort(grades.begin(), grades.end());
67     int mode=0, max_count=0, count=1;
68
69     for (size_t i = 1; i < grades.size(); ++i)
70     {
71         if (grades[i] == grades[i - 1])
72         {++count;}
73         else
74         {count = 1;}

```

```

75
76     if (count > max_count)
77     {max_count = count;
78     mode = grades[i];}
79     }
80
81     cout<<"Mode = "<<mode<<endl;
82
83     //e. Display the names of the students with the mode as their grade.
84     cout<<"\nStudents with the mode as their grade: ";
85     for (size_t i = 0; i < grades.size(); ++i)
86     {if (grades[i] == mode)
87     {cout<<names[grades.size()-(i+1)]<<" ";}
88     }
89     cout<<endl;
90
91     return 0;
92 }

```

```

C:\Users\SA\Downloads\ME-15-Sec-A\1st Semester\Courses\
Enter the number of name/grade pairs: 3
Enter name #1: A
Enter grade #1: 100
Enter name #2: B
Enter grade #2: 100
Enter name #3: C
Enter grade #3: 90
Mean = 96.6667
Median = 100
Mode = 100
Students with the mode as their grade: B A
-----
Process exited after 19.4 seconds with return value 0
Press any key to continue . . .

```

### **Task #3**

/\*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.\*/

```
#include<iostream>
```

```
#include<cmath>
```

```
using namespace std;
```

```
//Class named "Triangle"
```

```
class Triangle
```

```
{
```

```
    //Access specifier
```

```
    public:
```

```
    //Function to print area
```

```
    void area()
```

```
    {
```

```
        //Declaring variables
```

```
        int a=3, b=4, c=5, s, area;
```

```

        //Computing and displaying result
        s = (a+b+c)/2;
        area = sqrt(s*(s-a)*(s-b)*(s-c));
        cout<<"The Area of triangle is "<<area<<" square metres.\n"<<endl;
    }

//Function to print perimeter
void perimeter()
{
    //Declaring variables
    int base=3, perp=4, height=5, perimeter;

    //Computing and displaying result
    perimeter = base + perp + height;
    cout<<"The Perimeter of triangle is "<<perimeter<<"m.\n"<<endl;
}

};

int main()
{
    //Declaring an object of class "Triangle"

```

Triangle obj1;

//Accessing member functions

obj1.area();

obj1.perimeter();

}

```
1  /*3. Write a program to print the area and perimeter of a triangle having sides of 3 m, 4 m and 5 m
2     by creating a class named 'Triangle' with a function to print the area and perimeter.*/
3
4  #include<iostream>
5  #include<cmath>
6  using namespace std;
7
8
9  //Class named "Triangle"
10 class Triangle
11 {
12     //Access specifier
13     public:
14
15     //Function to print area
16     void area()
17     {
18         //Declaring variables
19         int a=3, b=4, c=5, s, area;
20
21         //Computing and displaying result
22         s = (a+b+c)/2;
23         area = sqrt(s*(s-a)*(s-b)*(s-c));
24         cout<<"The Area of triangle is "<<area<<" square metres.\n"<<endl;
25     }
26
27     //Function to print perimeter
28     void perimeter()
29     {
30         //Declaring variables
31         int base=3, perp=4, height=5, perimeter;
32
33         //Computing and displaying result
34         perimeter = base + perp + height;
35         cout<<"The Perimeter of triangle is "<<perimeter<<"m.\n"<<endl;
36     }
37 };
38
39
40 int main()
41 {
42     //Declaring an object of class "Triangle"
43     Triangle obj1;
44
45     //Accessing member functions
46     obj1.area();
47     obj1.perimeter();
48 }
```

C:\Users\SA\Downloads\ME-15-Sec-A\1st Semester\Courses\Com

The Area of triangle is 6 square metres.

The Perimeter of triangle is 12m.

---

Process exited after 0.07365 seconds with return value 0  
Press any key to continue . . .

## **Task #4**

/\*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.\*/\*

```
#include <iostream>
```

```
#include <vector>
```

```
using namespace std;
```

```
//Structure named "Company"
```

```
struct Company
```

```
{
```

```
    //Declaring data members
```

```
    string name;
```

```
    float salary;
```

```
    float hours_worked;
```

```
//Function to increase salary based on working hours
```

```
void increase_in_salary()
```

```
{
```

```
        //Increase salary by $50 for 8 hours worked
        if (hours_worked>=8&&hours_worked<10)
        {salary += 50;}

        //Increase salary by $100 for 10 hours worked
        else if (hours_worked>=10&&hours_worked<12)
        {salary += 100;}

        //Increase salary by $150 for 12 or more hours worked
        else if (hours_worked>=12)
        {salary += 150;}
    }
};
```

```
int main()
{
    //Declaring variables
    const int num_employee=10;
    Company employee[num_employee];

    //Taking inputs and computing result
    for (int i=0; i<num_employee; i++)
```



```

    {
    cout<<"Enter employee name: ";
    cin>>employee[i].name;
    cout<<"Enter employee salary ($): ";
    cin>>employee[i].salary;
    cout<<"Enter hours worked per day: ";
    cin>>employee[i].hours_worked;
    cout<<endl;
    employee[i].increase_in_salary();
    }

//Displaying result
cout<<"\nFinal Employee Information:"<<endl;
cout<<"\nName\t"<<"Final Salary"<<endl;
for (int i=0; i<num_employee; i++)
{
    cout<<" "<<employee[i].name<<"\t "<<employee[i].salary<<endl;
    }
return 0;
}

```

```

1  /*4. Write a structure to store the names, salary, and hours of work per day of 10 employees in a company.
2     Write a program to increase the salary depending on the number of hours of work per day as follows and
3     then print the name of all the employees along with their final salaries.*/
4
5  #include <iostream>
6  #include <vector>
7  using namespace std;
8
9  //Structure named "Company"
10 struct Company
11 {
12     //Declaring data members
13     string name;
14     float salary;
15     float hours_worked;
16
17     //Function to increase salary based on working hours
18     void increase_in_salary()
19     {
20         //Increase salary by $50 for 8 hours worked
21         if (hours_worked>=8&&hours_worked<10)
22             {salary += 50;}
23
24         //Increase salary by $100 for 10 hours worked
25         else if (hours_worked>=10&&hours_worked<12)
26             {salary += 100;}
27
28         //Increase salary by $150 for 12 or more hours worked
29         else if (hours_worked>=12)
30             {salary += 150;}
31     }
32 };
33
34

```

```

35 int main()
36 {
37     //Declaring variables
38     const int num_employee=10;
39     Company employee[num_employee];
40
41     //Taking inputs and computing result
42     for (int i=0; i<num_employee; i++)
43     {
44         cout<<"Enter employee name: ";
45         cin>>employee[i].name;
46         cout<<"Enter employee salary ($): ";
47         cin>>employee[i].salary;
48         cout<<"Enter hours worked per day: ";
49         cin>>employee[i].hours_worked;
50         cout<<endl;
51         employee[i].increase_in_salary();
52     }
53
54     //Displaying result
55     cout<<"\nFinal Employee Information:"<<endl;
56     cout<<"\nName\t"<<"Final Salary"<<endl;
57     for (int i=0; i<num_employee; i++)
58     {
59         cout<<" " <<employee[i].name<<"\t " <<employee[i].salary<<endl;
60     }
61     return 0;
62 }

```

C:\Users\SA\Downloads\ME-15-Sec-A\1st Semester\Courses\Co

Enter employee name: A  
Enter employee salary (\$): 100  
Enter hours worked per day: 6

Enter employee name: B  
Enter employee salary (\$): 100  
Enter hours worked per day: 7

Enter employee name: C  
Enter employee salary (\$): 100  
Enter hours worked per day: 8

Enter employee name: D  
Enter employee salary (\$): 100  
Enter hours worked per day: 9

Enter employee name: E  
Enter employee salary (\$): 100  
Enter hours worked per day: 10

Enter employee name: F  
Enter employee salary (\$): 100  
Enter hours worked per day: 11

Enter employee name: G  
Enter employee salary (\$): 100  
Enter hours worked per day: 12

Enter employee name: H  
Enter employee salary (\$): 100  
Enter hours worked per day: 13

Enter employee name: I  
Enter employee salary (\$): 100  
Enter hours worked per day: 14

Enter employee name: J  
Enter employee salary (\$): 100  
Enter hours worked per day: 15

Final Employee Information:

Name	Final Salary
A	100
B	100
C	150
D	150
E	200
F	200
G	250
H	250
I	250
J	250

---

Process exited after 69.01 seconds with return value 0  
Press any key to continue . . .