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<<" ";
    }
}
```

```cpp
/*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<<" ";
    }
}
```

```
/tmp/hVOMejaplL.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;}
```

```cpp
/*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;
```

```cpp
    //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;}
```

```
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

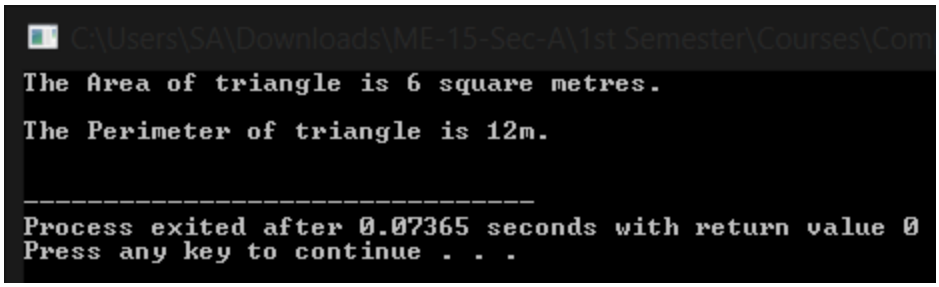
by creating a class named 'Triangle' with a function to print the area and perimeter.*/

```
#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();
}
```

```cpp
/*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();
}
```


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;
}
```

```cpp
/*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;
}
```


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 . . .