|  |  |  |
| --- | --- | --- |
| **Rubrics for Object Oriented Programming Lab** | | |
| **Lab #:** | **01** | |
| **Lab Title:** | **Introduction to C++ and Review of the Basic Programming Concept** | |
| **Submitted by:** | | |
| **Name** | | **Registration #** |
| **AMMAR**  **MUHAMMAD KALEEM ULLAH** | | **FA19-BCE-001**  **FA19-BCE-007** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Rubrics name & number** | | | **Marks** | | |
| **In-Lab** | | **Post-Lab** |
| **Engineering Knowledge** | ***R2: Use of Engineering Knowledge and follow Experiment Procedures:***  *Ability to follow experimental procedures, control variables, and record procedural steps on lab report.* | |  | | |
| **Problem Analysis** | | ***R5: Data/Evidence Measurements:***  *Ability to record raw data / evidence.* | |  | |
| **Design** | | ***R8: Best Coding Standards:***  *Ability to follow the coding standards and programming practices.* | |  | |
| **Modern Tools Usage** | | ***R9: Understand Tools:*** *Ability to describe and explain the principles behind and applicability of engineering tools.* | |  | |
| **Individual and Teamwork** | | ***R12: Individual Work Contributions:*** *Ability to carry out individual responsibilities.* | |  | |
| ***R13: Management of Team Work:***  *Ability to appreciate, understand and work with multidisciplinary team members.* | |  | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Rubrics #** | R2 | R5 | R8 | R9 | R12 | R13 |
| **In –Lab** |  |  |  |  |  |  |
| **Post- Lab** |  |  |  |  |  |  |

**Lab#01**

**Introduction to C++ and Review of Basic Programming Concepts**

1. **Objectives:**

Objectives of this lab are:

* Review basic programming concepts
* Understanding the C++ syntax
* Getting familiar with the concepts of structures
* Usage of Code Blocks IDE

1. **Introduction:**

C++ is a general-purpose programming language that supports various computer programming models such as object-oriented programming and generic programming. It was created by Bjarne Stroustrup and,

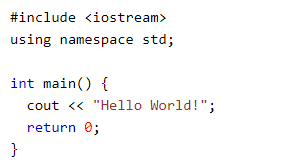
***“Its main purpose was to make writing good programs easier and more pleasant for the individual programmer.”***

By learning C++, you can create applications that will run on a wide variety of hardware platforms such as personal computers running Windows, Linux, UNIX, and Mac OS X, as well as small form factor hardware such as IoT devices like the Raspberry PI and Arduino–based boards.

C++ is derived from the C-Language. Almost every correct statement in C is also correct in C++, although the reverse is not true.

* 1. **Syntax of C++:**

**Basic program:**



**Explanation:**

**Line 1:** **#include <iostream>** is a header file library that lets us work with input and output objects, such as cout (used in line 5). Header files add functionality to C++ programs.

**Line 2:** **using namespace std** means that we can use names for objects and variables from the standard library.

**Line 3:** A blank line. C++ ignores white space.

**Line 4:** Another thing that always appear in a C++ program, is **int main()**. This is called a function. Any code inside its **curly brackets {}** will be executed.

**Line 5:** **cout** (pronounced "see-out") is an object used together with the **insertion operator (<<)** to output/print text. In our example it will output "Hello World".

**Line 6:** **return 0** ends the main function.

**Note:** Every C++ statement ends with a semicolon;

* 1. **Difference between C and C++:**

The **main difference** between both these languages is C is a procedural programming language and does not support classes and objects, while C++ is a combination of both procedural and object-oriented programming languages.

In **procedural programming**, each statement in the language tells the computer to do something: get some input, add these numbers divide by six, display that output. A program in procedural language is a list of instructions.

For every small program, no other organizing principle is needed. The programmer creates the list of instructions and the computer carries them out.

In **object – oriented programming** is to combine into a single unit both data and the function that operate on that data. Such a unit is called an object. Keep in mind that object – oriented programming is not primary concerned with the details of program operation, instead it deals with the overall organization of the program.

|  |  |  |
| --- | --- | --- |
| **Feature** | **C** | **C++** |
| Language Type | As mentioned before C is procedural programming. | On the other hand, C++ supports both procedural and object-oriented programming paradigms. |
| OOPs feature Support | As C does not support the OOPs concept so it has no support for polymorphism, encapsulation, and inheritance. | C++ has support for polymorphism, encapsulation, and inheritance as it is being an object-oriented programming language |
| Data Security | As C does not support encapsulation so data behave as a free entity and can be manipulated by outside code. | On another hand in the case of C++ encapsulation hides the data to ensure that data structures and operators are used as intended. |
| Driven type | C in general known as function-driven language. | On the other hand, C++ is known as object driven language. |
| Feature supported | C does not support function and operator overloading also do not have namespace feature and reference variable functionality. | On the other hand, C++ supports both function and operator overloading also have namespace feature and reference variable functionality. |

* 1. **Difference between Syntax of C and C++:**
* **Namespace** is used by C++, which avoid name collisions.
* **Header file** stdio.h in C, iostream.h in C++
* **scanf() and printf()** functions are used for input/output in C, **cin and cout** are used for input/output in C++.
  1. **Structure:**

A structure is a user defined data type. Through structures you have the ability to define a new type of data considerably more complex than the types we have been using. A structure is a combination of several different data types. It is similar to a class in that, both holds a collection of data of different data types. It is declared by using the keyword struct followed by the structure name.

**Syntax:**

struct struct\_name

{

Data\_type1 member\_name1;

Data\_type2 member\_name2;

Data\_type3 member\_name3;

} object\_name;

* 1. **Passing arguments by Value:**

By definition, pass by value means you are **making a copy in memory of the actual parameter's value that is passed** in, a copy of the contents of the actual parameter.

A parameter passing mechanism in which the value of actual parameter is copied to formal parameters of called functions is known as pass by value. If the function makes any change in formal parameter, it does not affect the values of actual parameter. **It is the default mechanism for passing parameters to functions.**

**Actual Parameters** are the values that are passed to the function when it is invoked while **Formal Parameters** are the variables defined by the function that receives values when the function is called.

* 1. **Passing arguments by Reference:**

A parameter passing mechanism in which the address of actual parameter is passed to the called function is known as **pass by reference (also called pass by address**). In pass by reference, we declare the function parameters as references rather than normal variables. The formal parameter is not created separately in the memory. Formal parameter becomes a second name of actual parameter. It means that single memory is shared between actual parameter and formal parameter. If the called function makes any change in formal parameter, the change is also visible in actual parameter.

**Difference:**

|  |  |
| --- | --- |
| **Pass by value** | **Pass by reference** |
| A copy of variable is created and changes are not reflected in original memory address | Instead of creating a copy of variable changes are made at original memory address |

1. **In-Lab Tasks:**
   1. **Task#01:** Write a program that declares a structure to store date. Declare an instance of this structure to represent date of birth. The program should read the day, month and year values of birth date and display date of birth in dd/mm/yy format.

* **Code:**

#include<iostream>

using namespace std;

/\*Structure to represent date of birth\*/

struct data

{

int day,month,year;

};

/\*Main Program\*/

int main()

{

data d; /\*Creating Object for the structure\*/

/\*Getting Data from the user using structure\*/

cout<<"Enter your day of birth : ";

cin>>d.day;

cout<<endl<<"Enter your month of birth : ";

cin>>d.month;

cout<<endl<<"Enter your year of birth : ";

cin>>d.year;

/\*Printing Data Acoording to the given requirments\*/

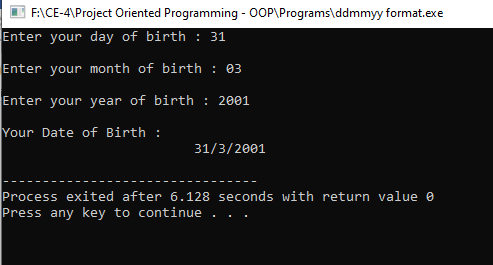
cout<<endl<<"Your Date of Birth : ";

cout<<endl<<"\t\t\t"<<d.day<<"/"<<d.month<<"/"<<d.year<<endl;

return 0;

}

* **Output:**

****

* 1. **Task#02:** Write a program that declares a structure to store Student data containing his name, age and Roll#. Use array of structures to represent record of 3 students.
* **Code:**

#include<iostream>

using namespace std;

/\*Structure to get data\*/

struct data

{

char name[100];

int age,roll;

};

//main function

int main()

{

data d[3]; //creating object in array to get data of students

/\*using loop to get data of three students\*/

for(int i=0;i<3;i++)

{

cout<<endl<<" Enter Number "<<i+1<<" Student Data : ";

cout<<endl<<" Enter the name : ";

cin>>d[i].name;

cout<<endl<<" enter the age : ";

cin>>d[i].age;

cout<<endl<<" Enter the roll number : ";

cin>>d[i].roll;

}

system ("CLS"); //a statement to clear the screen

/\*Printing Collected Record\*/

cout<<endl<<"------ Collected Data ---------";

for(int i=0;i<3;i++)

{

cout<<endl<<"Student Number "<<i+1<<"Data";

cout<<endl<<"Name: "<<d[i].name;

cout<<endl<<"Age: "<<d[i].age;

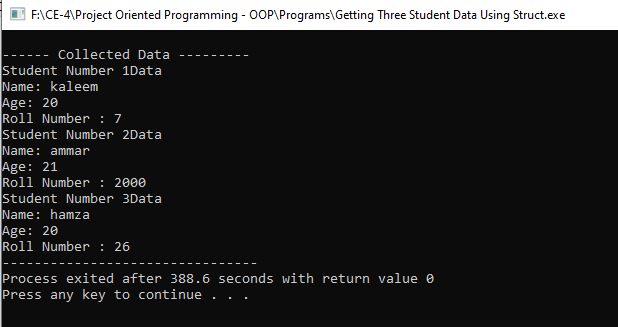
cout<<endl<<"Roll Number : "<<d[i].roll;

}

return 0;

}

* **Output:**

****

* 1. **Task#03:** Write a program that declares a structure to store book name, price and pages of a book. The structure should include functions to assign user defined values to each book and display the record of most costly book.
* **Code:**

#include<iostream>

using namespace std;

/\*Structure to Store data\*/

struct lib

{

char name[100];

int price,pages;

};

int main()

{

/\*Declaring varible and getting the number of book user want to enter\*/

int n;

cout<<"How many books data you want to enter :";

cin>>n;

lib l[n]; //creating object

/\*using for loop to get data from the user\*/

for(int i=0;i<n;i++)

{

cout<<endl<<"Enter book number "<<i+1<<" Data :";

cout<<endl<<"Enter the book Name: ";

cin>>l[i].name;

cout<<endl<<"Enter book price: ";

cin>>l[i].price;

cout<<endl<<"Enter book pages :";

cin>>l[i].pages;

}

/\*swapping the highest cost book's data to the end of the array\*/

for(int i=0;i<n;i++)

{

if(l[i].price>l[i+1].price)

{

swap(l[i].name,l[i+1].name);

swap(l[i].pages,l[i+1].pages);

swap(l[i].price,l[i+1].price);

}

}

/\*printing the last book data from array\*/

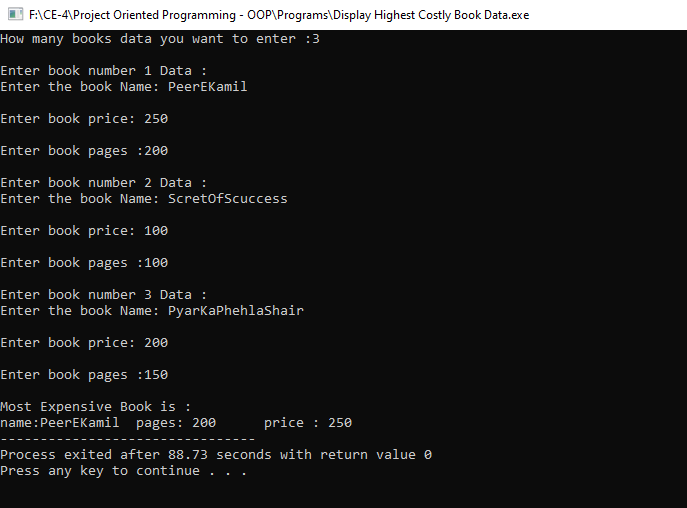
cout<<endl<<"Most Expensive Book is : ";

cout<<endl<<"name:"<<l[n-1].name<<"\t pages: "<<l[n-1].pages<<"\t price : "<<l[n-1].price;

return 0;

}

* **Output:**

****

* 1. **Task#04:** Write a function that swaps the values of two integer variables
     1. using pass by value
     2. and pass by reference and see their differences
* **Code:**

#include<iostream>

using namespace std;

//Function which swap value by value

void swapValue(int a,int b)

{

int temp;

temp=a;

a=b;

b=a;

}

//Function which swap a and b value using address

void swapAddress(int &a,int &b)

{

int temp;

temp=a;

a=b;

b=temp;

}

int main()

{

//declaring variables and getting data

int a,b,c;

cout<<"Enter the 'a'' value : ";

cin>>a;

cout<<endl<<"Enter the 'b' value : ";

cin>>b;

//getting option from the user to swap by value or by address

cout<<endl<<"How you want to swap ?";

cout<<endl<<" \n Enter '0' to swap by value and enter '1' to swap by reference .. :";

cout<<endl<<"Enter Your Choice : ";

cin>>c;

//swapping by value

if (c==0)

{

swapValue(a,b);

cout<<"\n a = "<<a;

cout<<"\n b = "<<b;

}

//swapping by address

else if (c==1)

{

swapAddress(a,b);

cout<<"\n a = "<<a;

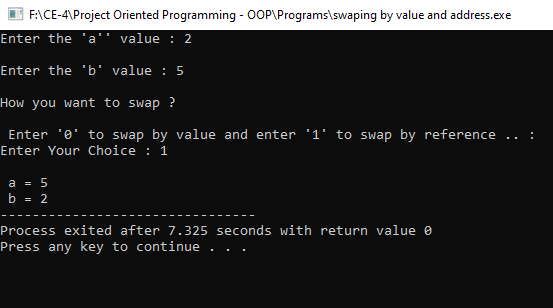
cout<<"\n b = "<<b;

}

return 0;

}

* **Output:**

****

1. **Post-Lab Tasks:**
   1. **Task#1:** There is a structure called **employee** that holds information like employee code, name, date of joining. Write a program to create an array of the structure and enter some data into it. Then ask the user to enter current date. Display the names of those employees whose tenure is 3 or more than 3 years according to the given current date

* **Code:**

#include<iostream>

using namespace std;

/\*Structure to get the exact dates\*/

struct Date

{

int date,month,year;

};

/\*Structure to get code and name\*/

struct employee

{

int code;

char name[50];

};

int main()

{

/\*varible declaration to know how many employee's data user wants to enter\*/

int n;

cout<<"How many Employee's Data You want to enter:";

cin>>n;

/\*creating objects\*/

employee e[n];

Date doj[n]; //doj=date of joining

Date currentDate;

/\*getting employee's data from user'\*/

for(int i=0;i<n;i++)

{

cout<<"Enter The Number "<<i+1<<" Employee's Data .... \n";

cout<<"Enter the Name: ";

cin>>e[i].name;

cout<<endl<<"Enter the employee code: ";

cin>>e[i].code;

cout<<endl<<"Enter the Date of Joining: ";

cout<<endl<<"Enter the Date : ";

cin>>doj[i].date;

cout<<endl<<"Enter the Month: ";

cin>>doj[i].month;

cout<<endl<<"Enter the Year: ";

cin>>doj[i].year;

}

/\*getting the current date\*/

cout<<endl<<"Enter the Current Date: ";

cout<<endl<<"date : ";

cin>>currentDate.date;

cout<<"Month : ";

cin>>currentDate.month;

cout<<"Year : ";

cin>>currentDate.year;

/\*showing the employee's data who's tensure is 3 or more than 3 years \*/

cout<<endl<<"Employee having tenure 3 or more than three year ... \n";

for(int i=0;i<n;i++)

{

if((currentDate.year-doj[i].year)>=3)

{

cout<<endl<<"Employee Name : "<<e[i].name;

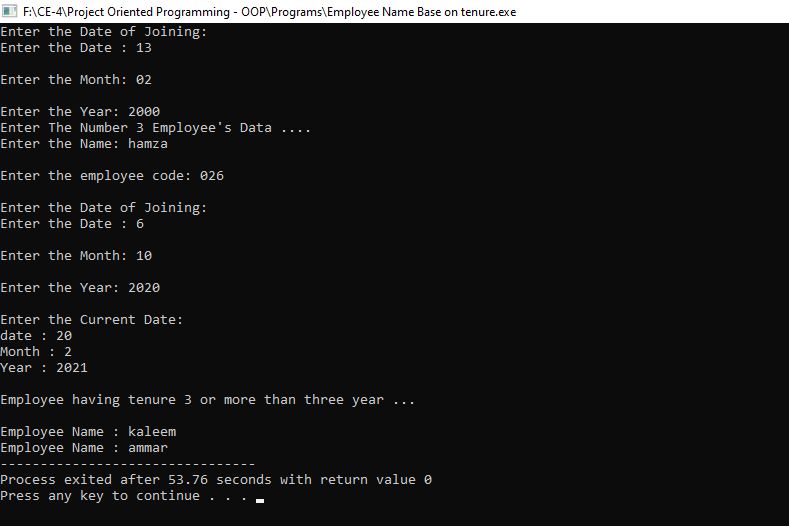
}

}

return 0;

}

* **Output:**

****

1. **Conclusion:**

In this lab we have seen the importance and use of the structure in C++

programming language, go through the basic syntax of the C++ whereas structures and practice question related to the implementation of the structures so,

After completing this Lab, we are now able to implement code using structures.