Object Oriented Programming

Session: Fall 2022

Faculty of Information Technology
UCP Lahore, Pakistan

Lab 1: Revision Lab

Before we move to Object Oriented Programming in its true sense, we will today revisit C++ Structures which are closely related to classes. This lab session is intended to give a recap of:

- Usage of a structure.
- Syntax of defining a structure and accessing it members.
- Using pointers to structures.
- Nested structures.

A structure is a group of data elements grouped together under one name. These data elements, known as *members*, can have different types and different lengths. Data structures are declared in C++ using the following syntax:

```
struct structure_name {
member_type1 member_name1;
member_type2 member_name2;
member_type3 member_name3;
.
.
} object_names;
```

where structure_name is a name for the structure type, object_name can be a set of valid identifiers for objects that have the type of this structure. Within braces { } there is a list with the data members, each one is specified with a type and a valid identifier as its name.

The first thing we have to know is that a data structure creates a new type: Once a data structure is declared, a new type with the identifier specified as structure_name is created and can be used in the rest of the program as if it was any other type. For example:

```
struct product {
  int weight;
  float price;
} ;

product apple;
product banana, melon;
```

```
struct product {
  int weight;
  float price;
} apple, banana, melon;
```

```
apple.weight
apple.price
banana.weight
banana.price
melon.weight
melon.price
```

Struct with functions

```
// example about structures
#include <iostream>
#include <string>
using namespace std;

struct Movies {
  string title;
  int year;
};

void printmovie (Movies movie);

int main ()
{
  Movies mine, yours;

  mine.title = "2001 A Space Odyssey";
  mine.year = 1968;

  cout << "Enter title: ";
  cin >> yours.title;
  cout << "Enter year: ";
  cin  >> yours.year;

  cout << "My favorite movie is:\n ";
  printmovie (mine);
  cout << "And yours is:\n ";
  printmovie (yours);
  return 0;
}

void printmovie (Movies movie)
{
  cout << movie.title <<endl;
  cout <<  movie.year << endl;
}
```

```
Enter title: Alien
Enter year: 1979

My favorite movie is:
 2001 A Space Odyssey (1968)
And yours is:
 Alien (1979)
```

Array of Structures

```
// array of structures
#include <iostream>
#include <string>
using namespace std;

const int NUM = 3;

struct Movies {
  string title;
  int year;
};

void printmovie (Movies movie);

int main ()
{

 Movies films [NUM]
 int n;
  for (n=0; n<NUM; n++)
  {
    cout << "Enter title: ";
    cin >> films[n].title);
    cout << "Enter year: ";
    cin >> films[n].year;
   }

  cout << "\nYou have entered these
movies:\n";
  for (n=0; n<NUM; n++)
    printmovie (films[n]);
  return 0;
}

void printmovie (Movies movie)
{
  cout << movie.title << endl;
  cout << movie.year << endl;
}
```

```
Enter title: Blade Runner
Enter year: 1982
Enter title: Matrix
Enter year: 1999
Enter title: Taxi Driver
Enter year: 1976

You have entered these movies:
Blade Runner (1982)
Matrix (1999)
Taxi Driver (1976)
```

Pointer to Structure

```
// pointers to structures
#include <iostream>
#include <string>
using namespace std;

struct Movies {
  string title;
  int year;
};

int main ()
{
  Movies amovie;
  Movies * pmovie;
  pmovie = &amovie;

  cout << "Enter title: ";
  cin >> pmovie->title;
  cout << "Enter year: ";
  cin >> pmovie->year;

  cout << "\nYou have entered:\n";
  cout << pmovie->title <<endl;
  cout << pmovie->year << endl;

  return 0;
}
```

```
Enter title: Invasion of the body
snatchers
Enter year: 1978

You have entered:
Invasion of the body snatchers
(1978)
```

Nested Struct

```
struct Movies {
  string title;
  int year;
};

struct Friends {
  string name;
  string email;
  Movies favorite_movie;
  };

Friends charlie, maria;
```

```
charlie.name
maria.favorite_movie.title
charlie.favorite_movie.year
```

Lab Objectives:

The basic purpose of this laboratory is revision of some concepts of c++ that has been covered in the course of Introduction to Computing and Programming Fundamentals. Its objective is to:

- Recall student's previously learned basic concepts.
- Revision of structures and pointers.
- Understanding problem statements and designing an appropriate solution.

Instructions:

- Indent your code
- Comment your code
- Use meaningful variable names
- Plan your code carefully on a piece of paper before you implement it.
- Name of the program should be same as the task name. i.e. the first program should be Task_1_01.cpp

Lab Tasks:

Lab Task 1:

Create a `struct` called Student which has registration no, name, admission date and cgpa as member variables. Create a dynamic **array** of type **Student**.

Write a function `Input_Records` for getting record of a student. Also write a function Display to print contents of the `struct` **student** on screen.

Lab Task 2:

Create a `struct` called UCP which contains further structs of with the name of department and department contains the structs of faculty, students, HR, admin.

1. Add faculty information including their id, names, designation, department, salary using input function and also display information of faculty using display function
2. Add student information including their roll no, names, cgpa, discipline, mobile and address as input function and also display information of faculty using display function
3. Add staff information of their id, names, designation, department, salary using input function and also display information of faculty using display function
4. Add HR employee, information including their id, names, designation, department, salary using input function and also display information of faculty using display function