Object-Oriented Programming Lab

SPRING - 2024

LAB 05

FAST National University of
Computer and Emerging Sciences

Learning Outcomes

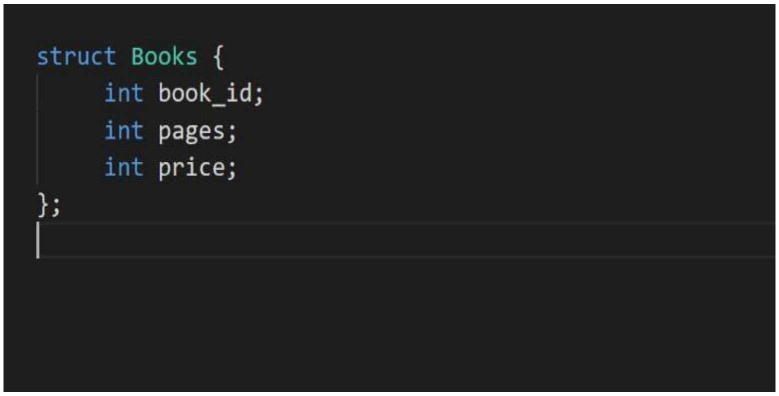
In this lab you are expected to learn the following:

- Structures
- Recursion

Structures

Structure is a user-defined data type that allows you to combine data items of different kinds.

Structure Declaration



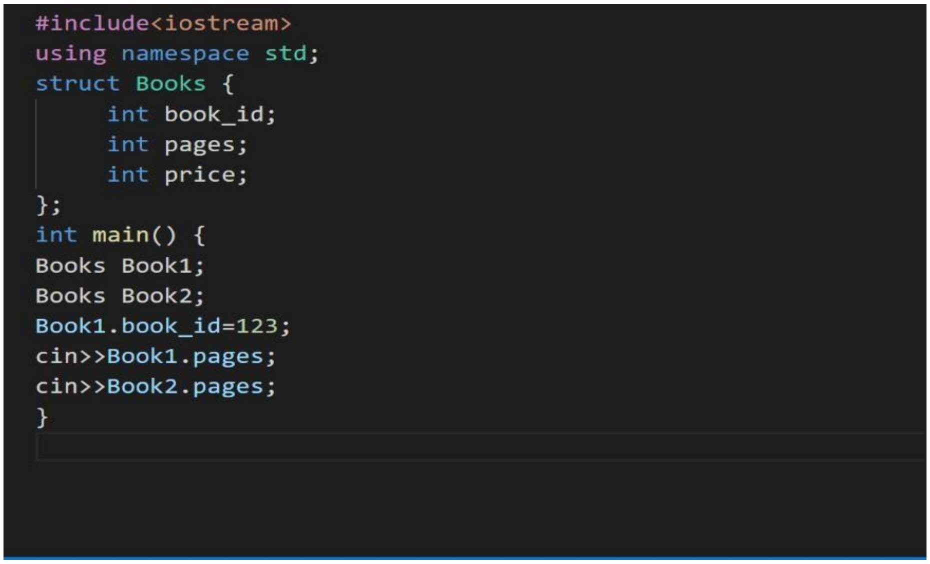
```
struct Books {
      int book_id;
      int pages;
      int price;
};
```

Types of Members

Structures in C++ can contain two types of members:

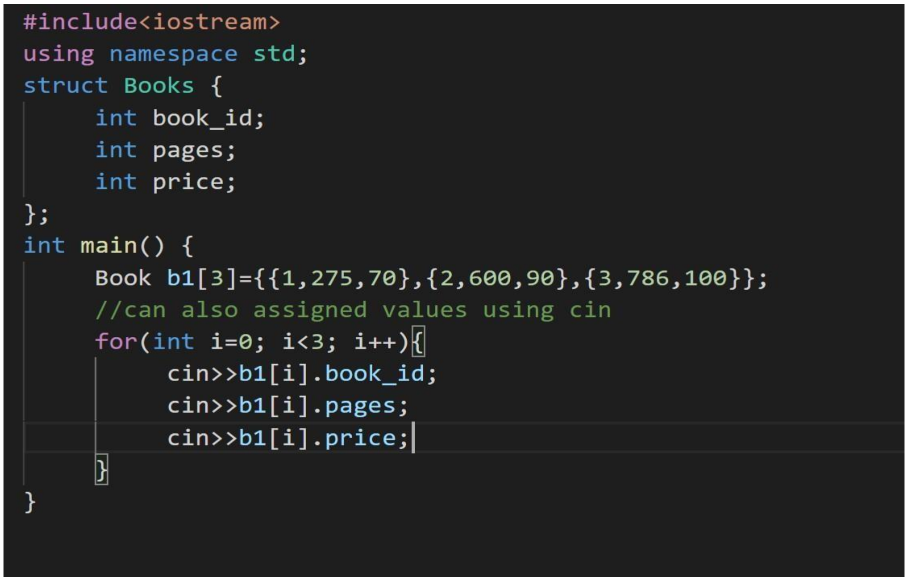
- **Data Member:** These members are normal C++ variables. We can create a structure with variables of different data types in C++.
- **Member Functions:** These members are normal C++ functions. Along with variables, we can also include functions inside a structure declaration.

Accessing Structure Members



```
#include<iostream>
using namespace std;
struct Books {
      int book_id;
      int pages;
      int price;
};
int main() {
Books Book1;
Books Book2;
Book1.book_id=123;
cin>>Book1.pages;
cin>>Book2.pages;
}
```

Array of Structures

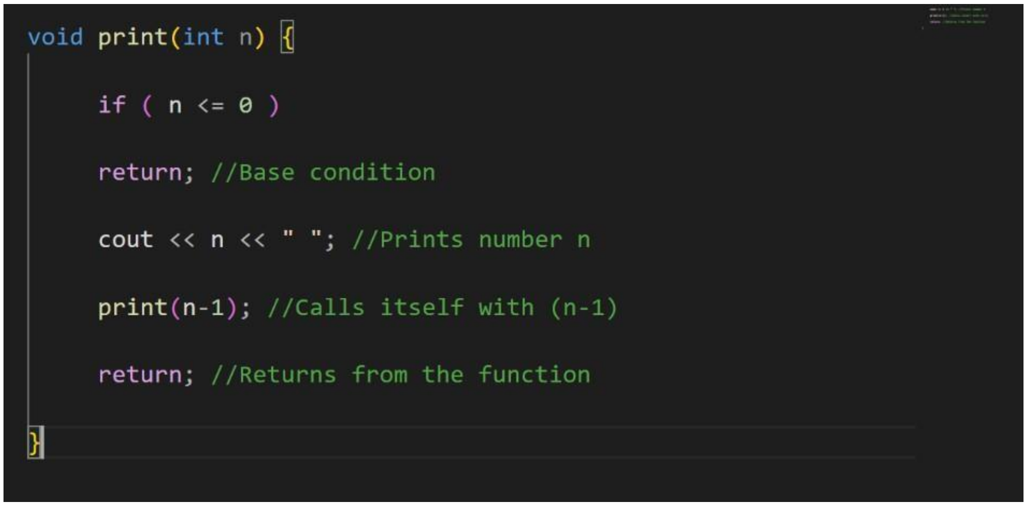


```
#include<iostream>
using namespace std;
struct Books {
	int book_id;
	int pages;
	int price;
};
int main() {
	Book b1[3]={{1,275,70},{2,600,90},{3,786,100}};
	//can also assigned values using cin
	for(int i=0; i<3; i++){
		cin>>b1[i].book_id;
		cin>>b1[i].pages;
		cin>>b1[i].price;
	}
}
```

RECURSION

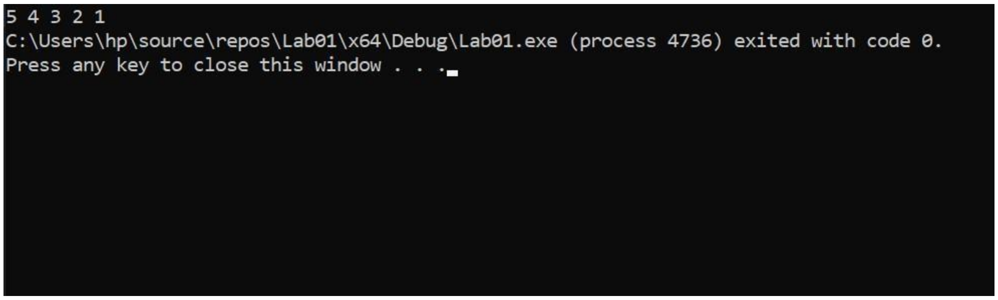
A recursive function is one that calls itself.

Example



```
void print(int n) {

    if ( n <= 0 )

    return; //Base condition

    cout << n << " "; //Prints number n

    print(n-1); //Calls itself with (n-1)

    return; //Returns from the function

}
```

Output



```
5 4 3 2 1
C:\Users\hp\source\repos\Lab01\x64\Debug\Lab01.exe (process 4736) exited with code 0.
Press any key to close this window . . .
```

Lab Task

Submission Instructions:

1. Create a single .h file containing all the functions of the problems.
2. Now create a new folder with name *ROLLNO_SEC_LAB05* **e.g. i22XXXX_A_LAB05**
3. Move all of your files to this newly created directory and compress it into a **.zip file**.
4. Now you have to submit this zipped file on Google Classroom.
5. Plagiarism in the Lab Task will result in **zero** marks in the whole category.

Question 1.

Write a recursive function to reverse a string without reversing the characters of words in string. Example: “this is a car” becomes “car a is this”.

Prototype: string reverse_string(string s)

Question 2.

A. Declare a structure named Complex having two data members named;
   - real of type int
   - imaginary of type int.

B. Create a member function addComplex that takes two arguments, the first argument should be passed by value and the second should be passed by reference. Your addComplex function should add two complex numbers using structures. To add two complex numbers, just add the corresponding real and imaginary parts. For Example, the sum of 5 + 3*i* and 4 + 2*i* is 9 + 5*i.*

Prototype: Complex addComplex(Complex c1, Complex &c2);

Question 3.

A. Declare a structure Named Student to store the information of 5 students taken from the user. The Data members should be:
   - Name of type string
   - roll_no of type string
   - age of type int

B. Write a member function getNames which takes a dynamic array of type Student of size 5 and will return the names of all students having age 18.

Prototype: string* getNames(Student *std);

C. Write a member function getEvenRollno which takes a dynamic array of type Student of size 5 and will return information of all students having even roll numbers. In order to get an even roll number just check its last digit.

Prototype: Student* getEvenRollno(Student *std);

Question 4.

A. Declare a structure named as CustomTime having three data members named;

- hours of type int
- mins of type int
- secs of type int

B. Write a member function timeToSecs which takes CustomTime object t1 as a function parameter, Calculate the total seconds in time and convert this object t1 to seconds (of typeint) and return it.

Prototype: int timeToSecs(CustomTime t1);

C. Write a member function AddTimes which takes CustomTime object t1 and CustomTimeobject t2 as parameters.

1. You have to convert these two objects t1, t2 in seconds using above-definedfunction timeToSecs.
2. Add both seconds returned from above step.

Prototype: int AddTimes(CustomTime t1, CustomTime t2);

Question 5.

A. Declare a structure named as CourseRegistration having data members named;

- courseCode of type string
- courseTitle of type string
- CreditHours of type int
- Section of type char

B. Declare a structure named as SemesterRegistration having data members named;

- semesterCode of type string
- course_reg of type CourseRegistration (a dynamic array of size 5)

C. Write a member function GetCreditHoursCount which takes SemesterRegistration object s as parameter and return total numbers of credit hours registered in it.

Prototype: int GetCreditHoursCount(semesterRegistration s);

D. Write a member function FindCourseInSemesterRegistration which takes SemesterRegistration object s and a course code as parameter and return true if the courseis registered in the semester.

Prototype: bool FindCourseInSemesterRegistration(SemesterRegistration p , string courseCode)