Structures

Lab Sections

1. Objectives
2. Introduction
3. Definitions
4. Declaration Syntax
5. Experiments

Declarations of Structures

1. Objectives

After you complete this experiment you will be able to:

a. **determine the amount of memory a structure uses;**
b. **declare a structure;**
c. **discuss the properties of a structure.**

2. Introduction

A **structure** is a heterogeneous data type. We say it is heterogeneous because it may be composed of members that all have different types. The members of a structure are accessed using the dot operator, and all members are public by default.

3. Definitions

We will define several terms that you need to know to really understand structures. They are as follows:

a. The **dot operator** is used to access the members of a structure. The symbol, '**.**', is used to represent the dot operator.
b. The **total amount of memory** a structure uses is equal to the sum of all the memory (bytes) used by its members.
c. A structure is a **user defined data type**.
d. A structure is a **complex data type** because it may be composed of other structures and data types.
e. A structure is a heterogeneous data type because it may be composed of members that all have different types.

4. Declaration Syntax

To declare a structure:

```
struct structure_name
{
      field_type_1  field_name_1;
      . . . .
      field_type_n  field_name_n;
};
```

Notice that the member fields are enclosed in braces and that the right brace is followed by a semicolon.

More information on structures can be found in your course textbook and on the web.

5. Experiments

Step 1: In this experiment you will learn how to declare structures and access their members. Enter, save, compile and execute the following program in MSVS. Call the new project "StructuresExp1" and the program "structureDecls1.cpp". Answer the questions below:

```
#include <iostream>
#include <string>

using namespace std;

struct student_record
{
	string firstname, lastname;
	double age, income;
	int number_of_children;
	char sex;
};

int main()
{
	student_record Mary;

	cout<<"Enter the firstname and lastname: ";
	cin>>Mary.firstname;
	cin>>Mary.lastname;
	cout<<"Enter age: ";
	cin>>Mary.age;
	cout<<"Enter income: ";
	cin>>Mary.income;
	cout<<"Enter number of children: ";
	cin>>Mary.number_of_children;
	cout<<"Enter sex: ";
	cin>>Mary.sex;

	cout<<Mary.firstname<<"    "<<Mary.lastname<<endl;
	cout<<Mary.age<<endl;
	cout<<Mary.income<<endl;
	cout<<Mary.number_of_children<<endl;
	cout<<Mary.sex<<endl;

	return 0;
}
```

Question 1: Please discuss the output (if any) and any compiler errors or warnings?

Question 2: Please move the declaration of the structure "student_record" in the program in Step 1 inside the main program body (before the declaration "student_record Mary;"). Explain your observations.

Step 2: In this experiment you will learn how to declare structures and access their members. Enter, save, compile and execute the following program in MSVS. Call the new project "StructureExp2" and the program "StructureDecls2.cpp". Answer the questions below:

```
#include <iostream>
#include <string>

using namespace std;

struct student_record
{
	string firstname, lastname;
	double age, income;
	int number_of_children;
	char sex;
};

int main()
{

	student_record Mary;
	student_record Susan;

	cout<<"Enter the firstname and lastname: ";
	cin>>Mary.firstname;
	cin>>Mary.lastname;
	cout<<"Enter age: ";
	cin>>Mary.age;
	cout<<"Enter income: ";
	cin>>Mary.income;
	cout<<"Enter number of children: ";
	cin>>Mary.number_of_children;
	cout<<"Enter sex: ";
	cin>>Mary.sex;

	Susan = Mary;

	cout<<Susan.firstname<<"    "<<Mary.lastname<<endl;
	cout<<Susan.age<<endl;
	cout<<Susan.income<<endl;
	cout<<Susan.number_of_children<<endl;
	cout<<Susan.sex<<endl;

	return 0;
}
```

Question 3: What can you say about the assignment operator shown in the program in Step 2 when the left and right operands are structures?

Step 3: In this experiment you will investigate how the =, ==, <= and >= operators are used to compare two different structures. Enter, save, compile and execute the following program in MSVS. Call the new project "StructureLogOpsExp" and the program "StructureLogOps.cpp". Answer the questions below:

```
#include <iostream>
#include <string>

using namespace std;

struct student_record
{
	string firstname, lastname;
	double age, income;
	int number_of_children;
	char sex;
};

int main()
{

	student_record Mary;
	student_record Susan;

	cout<<"Enter the firstname and lastname: ";
	cin>>Mary.firstname;
	cin>>Mary.lastname;
	cout<<"Enter age: ";
	cin>>Mary.age;
	cout<<"Enter income: ";
	cin>>Mary.income;
	cout<<"Enter number of children: ";
	cin>>Mary.number_of_children;
	cout<<"Enter sex: ";
	cin>>Mary.sex;

	Susan = Mary;

	if (Susan == Mary)
	{
		cout<<Susan.firstname<<"    "<<Mary.lastname<<endl;
		cout<<Susan.age<<endl;
		cout<<Susan.income<<endl;
		cout<<Susan.number_of_children<<endl;
		cout<<Susan.sex<<endl;
	}
	return 0;
}
```

Question 4: What can you say about the logical equaility operator when the left and right operands are structures?

Question 5: What happens when you replace the logical equality operator (==) with the <= operator in the program in Step 3? Discuss your observations.

Question 6: What happens when you replace the logical equality operator (==) with the >= operator in the program in Step 3? Discuss your observations.

Question 7: What would you do to correct the program in Step 3, if necessary?