**Lab Sections**

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

|  |
| --- |
|  |

Array of Structures

**Array of Structures**

1. **Objectives**

**After you complete this experiment you will be able to implement and use an array of structures.**

1. **Introduction**

Arrays of structures are very easy to implement and use once you know how to implement as simple static array of integers. The operators all work the same. However, you must pay close attention to the subtle differences in the syntax needed.

1. **Definitions**

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

1. A structure is a heterogeneous data type.
2. The dot “.” operator is used to access the fields/members of a structure.
3. The name of a **static array** is a **constant** pointer to the first element in the array.
4. The **size/capacity** is the number of memory cells allocated to an array.
5. An **index /subscript** is used to access the memory cells in an array.
6. **[ ]** is called the subscript operator.
7. The **index** is a non-negative integer.
8. The **range** of an index is between 0 and the size-1.
9. **Declaration Syntax**

**a. To declare a structure:**

struct structure\_name

{

field\_type\_1 field\_name\_1;

. . . .

field\_type\_n field\_name\_n;

};

**Examples allocating memory for two Dynamic Arrays:**

struct student\_record

{

string firstname, lastname;

double age, income;

int number\_of\_children;

char sex;

};

**b. To declare an array of structures:**

structure\_name array\_of\_structures[size];

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

1. **Experiments**

**Step 1: In this experiment you will learn how to declare and use an array of structures.**

**Enter, save, compile and execute the following program in MSVS. Call the new project “StaticArraysExp1” and the program “StaticArrays1.cpp”. Answer the questions that follow.**

#include <iostream>

#include <string>

using namespace std;

struct student\_record

{

string name;

double age;

char sex;

};

int main()

{

student\_record Student\_DB[3] = {"Lofton",53,'M',"Thomas",55,'M',

"Tami",25,'F'};

cout<<endl;

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

{

cout<<Student\_DB[i].name<<" "

<<Student\_DB[i].age<<" "

<<Student\_DB[i].sex<<endl<<endl;

}

Student\_DB[0].name = "William";

Student\_DB[2].age = 100;

Student\_DB[1].sex = 'F';

cout<<"+++++++++++++++++++++++++++++++\n";

cout<<endl;

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

{

cout<<Student\_DB[i].name<<" "

<<Student\_DB[i].age<<" "

<<Student\_DB[i].sex<<endl<<endl;

}

return 0;

}

1. Please explain how the array “Student\_DB” was initialized?
2. What is the purpose of the two “for” loops in the program (StaticArray1s.cpp) in Step 1? Explain your answer.

**Step 2: In this experiment you will explain the output of a program that uses a dynamic array of structures. Enter, save, compile and execute the following program in MSVS. Call the new project “StaticArraysExp2” and the program “StaticArrays2.cpp”. Answer the questions that follow.**

#include <iostream>

#include <string>

using namespace std;

struct student\_record

{

string name;

double age;

char sex;

};

int main()

{

student\_record \*Student\_DB = new student\_record[3];

Student\_DB[0].name = "Lofton";

Student\_DB[0].age = 53;

Student\_DB[0].sex = 'M';

Student\_DB[1].name = "Thomas";

Student\_DB[1].age = 55;

Student\_DB[1].sex = 'M';

Student\_DB[2].name = "Tami";

Student\_DB[2].age = 25;

Student\_DB[2].sex = 'F';

cout<<endl;

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

{

cout<<Student\_DB[i].name<<" "

<<Student\_DB[i].age<<" "

<<Student\_DB[i].sex<<endl<<endl;

}

Student\_DB[0].name = "William";

Student\_DB[2].age = 100;

Student\_DB[1].sex = 'F';

cout<<"+++++++++++++++++++++++++++++++\n";

cout<<endl;

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

{

cout<<Student\_DB[i].name<<" "

<<Student\_DB[i].age<<" "

<<Student\_DB[i].sex<<endl<<endl;

}

delete [ ] Student\_DB;

return 0;

}

1. Please state and explain differences in the source code of the program (StaticArrays1.cpp) in Step 1 and the program (StaticArrays2.cpp) in Step 2?
2. What differences between static and dynamic arrays can you conclude from your observations?

**Step 3: In this experiment you will investigate the output of a program that uses static and dynamic arrays. Enter, save, compile and execute the following program in MSVS. Call the new project “StaticArraysExp3” and the program “StaticArrays3.cpp”. Answer the question that follows.**

#include <iostream>

using namespace std;

int main()

{

int static\_Array[5];

int \*dynamic\_Array;

dynamic\_Array = new int[5];

int i;

for(i=0; i<5; i++)

{

static\_Array[i]=i;

dynamic\_Array[i]=5;

}

for (i=0; i<5; i++)

{

cout<<"static\_Array["<<i<<"] = "<<static\_Array[i]<<endl;

cout<<"dynamic\_Array["<<i<<"] = "<<dynamic\_Array[i]<<endl;

}

cout<<endl<<endl<<endl;

dynamic\_Array = static\_Array;

for (i=0; i<5; i++)

{

cout<<"static\_Array["<<i<<"] = "<<static\_Array[i]<<endl;

cout<<"dynamic\_Array["<<i<<"] = "<<dynamic\_Array[i]<<endl;

}

return 0;

}

1. What action(s) does the program (StaticArrays3.cpp) in Step 3 perform?

**Step 4: In this experiment you will explain the output of a program that uses static and dynamic arrays. Enter, save, compile and execute the following program in MSVS. Call the new project “StaticArraysExp4” and the program “StaticArrays4.cpp”. Answer the question that follows.**

#include <iostream>

using namespace std;

int main()

{

int \*static\_Array = new int[5];

int \*dynamic\_Array;

dynamic\_Array = new int[5];

int i;

for(i=0; i<5; i++)

{

static\_Array[i]=i;

dynamic\_Array[i]=5;

}

for (i=0; i<5; i++)

{

cout<<"static\_Array["<<i<<"] = "<<static\_Array[i]<<endl;

cout<<"dynamic\_Array["<<i<<"] = "<<dynamic\_Array[i]<<endl;

}

cout<<endl<<endl<<endl;

static\_Array = dynamic\_Array;

for (i=0; i<5; i++)

{

cout<<"static\_Array["<<i<<"] = "<<static\_Array[i]<<endl;

cout<<"dynamic\_Array["<<i<<"] = "<<dynamic\_Array[i]<<endl;

}

return 0;

}

1. Compare the programs that were presented in Steps 1, 2, 3 and 4. What are your observations?

**Step 5: In this experiment you will explain the output of a program that uses static and dynamic arrays. Enter, save, compile and execute the following program in MSVS. Call the new project “StaticArraysExp5” and the program “StaticArrays5.cpp”. Answer the question that follows.**

#include <iostream>

using namespace std;

int main()

{

int static\_Array[5];

int \*dynamic\_Array;

dynamic\_Array = new int[5];

int i;

for(i=0; i<5; i++)

{

static\_Array[i]=i;

dynamic\_Array[i]=5;

}

for (i=0; i<5; i++)

{

cout<<"static\_Array["<<i<<"] = "<<static\_Array[i]<<endl;

cout<<"dynamic\_Array["<<i<<"] = "<<dynamic\_Array[i]<<endl;

}

cout<<endl<<endl<<endl;

static\_Array = dynamic\_Array;

for (i=0; i<5; i++)

{

cout<<"static\_Array["<<i<<"] = "<<static\_Array[i]<<endl;

cout<<"dynamic\_Array["<<i<<"] = "<<dynamic\_Array[i]<<endl;

}

return 0;

}

1. Why does the program in Step 5 produce compilation errors?