**LESSON SET 9**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Pointers**

**OBJECTIVES FOR STUDENT**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Lesson 9A:**

1. To learn about and use pointers

2. To learn about and use the concept of dynamic memory

**Lesson 9B:**

3. To work with pointers and arrays

4. To write complete programs using pointers

**ASSUMPTIONS**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Lesson 9A:**

1. Students understand and can work with arrays

2. Students understand dynamic variables

**Lesson 9B:**

1. Students understand the relationship between pointers and arrays as

discussed in the text and Pre-lab Reading Assignment

**PRE-LAB WRITING ASSIGNMENT SOLUTIONS**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

1. \*

2. &

3. pointer or reference

4. new

5. delete

6. reference

7. an address

8. 3.75

9. an address

10. 3.75

**LAB ASSIGNMENTS**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Lesson 9A:**

Lab 9.1: Introduction to pointer variables

Lab 9.2: Dynamic memory

**Lesson 9B:**

Lab 9.3: Dynamic arrays

Lab 9.4: Student generated code assignments

**LESSON 9A**

**LAB 9.1: Introduction to Pointer Variables**

Lab 9.1 gives students their first practice in working with pointers. Given length

and width, the program will calculate the area of a rectangle. This is a simple

process, however, it is done with the use of pointers.

A solved program is found in pointersKEY.cpp in the instructor’s folder for

Lesson Set 9.

**LAB 9.2: Dynamic Memory**

This program uses dynamic variables. It also tests the students’ knowledge of

pointers to character arrays. This can be a challenging assignment for students

learning pointers for the first time.

Students who do not understand that an array name represents the starting

address of the array may have difficulty in solving this code and answering the

questions in Exercise 2. In the cin and cout statements in the loop, use of the

dereferencing operator should be used \*(name +pos) even though name[pos]

works (which is dealt with in Exercise 2). Arrays and pointers are further discussed

in the next lab. This lab also tests the student’s knowledge of the new and delete

operators.

A solved program is found in dynamicKEY.cpp in the instructor’s folder for

Lesson Set 9.

*Answers to Exercise 2:*

For inputting and outputting the name, you were asked NOT to use a bracketed

subscript. Why is a bracketed subscript not necessary?

**The variable name is a pointer to the starting location of the array.**

**Any element can be accessed by giving this address plus the subscript.**

**Example:** \*(name+1) **gives the second element of the array.**

Would using name[pos] work for inputting the name? Why or why not?

**YES.** name[pos] **is the more common way of accessing elements of**

**the array and is perfectly legal!!**

Would using name[pos] work for outputting the name? Why or why not?

**YES.** name[pos1] **is the more common way of accessing elements of**

**the array and is perfectly legal!!**

**LESSON 9B**

**LAB 9.3: Dynamic Arrays**

This lab is very similar to Sample Problem 9.3 in the Pre-lab Reading Assignment.

It asks the students to allocate and deallocate memory for the array pointed to by

the defined pointer variable. The program uses input to determine the size of

the array. Students should be encouraged not to look back at the reading material

unless it is absolutely necessary.

A solved program is found in darrayKEY.cpp in the instructor’s folder for Lesson Set 9.

**LAB 9.4: Student Generated Code Assignments**

All three options require the students to generate dynamic arrays and pass them

as parameters.

The pre-lab reading material briefly covers dynamic arrays as parameters;

however, the student assignment explains that in fact they are passed as a pointer

variable. These exercises may be more challenging than earlier student generated

code exercises. Option 3 is probably the easiest of the three. Option

1 requires the use of a sort function while Option two uses a search.

*Option 1:* This program reads scores into a dynamically allocated array. Those

scores are sorted before being printed to the screen. The average score is

also found.

A solved program is found in scoretableKEY.cpp in the instructor’s folder for Lesson Set 9.

*Option 2:* This program reads id numbers into a dynamically allocated array. It

also reads an id number to be searched. A function determines where that

id number is in the array.

A solved program is found in searchKEY.cpp in the instructor’s folder for Lesson Set 9.

*Option 3:* This program asks that monthly sales be read into a dynamically allocated array. A function finds the yearly sales (sum of all the sales in the array). Another function finds the average of all the sales.

A solved program is found in yearlysalesKEY.cpp in the instructor’s folder for Lesson Set 9.

Possible solutions to all labs are given in the instructor’s folder for Lesson Set 9.