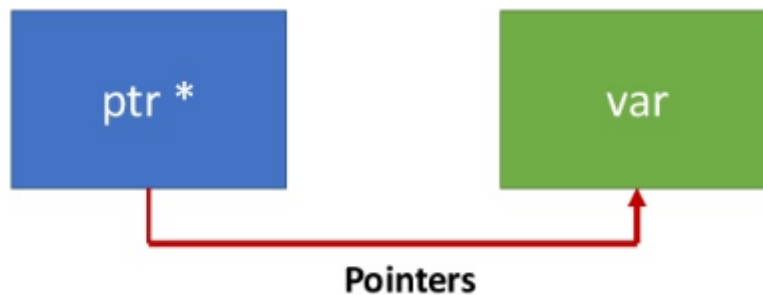




Tennessee
TECH

CSC 1300 LAB 9

BASIC POINTERS & VECTORS



CONCEPTS

- Basics of Pointers
- Vectors
- Selection Sort Algorithm

PAIRED PROGRAMMING OPTION

You may choose to select a lab partner for this lab, however, you **MUST** follow the instructions below if you do so. Failure to follow all the instructions for paired programming will result in a severe grade reduction for the assignment.

Your first step is to **exchange preferred contact information** just in case you are unable to complete the lab during lab and need to meet outside of lab class to finish.

SUBMISSION IN ILEARN

- **You will both upload the same exact zip file to your Lab 2 assignment in ilearn.**
- The text file should have both your names at the top and each source file should have both of your names in the comment block at the top.
- Both students will receive the same feedback and grade.

HOW TO PAIR PROGRAM

- **DO NOT DIVY UP THE PARTS AND WORK ON THEM SEPARATELY.**
- One of you can start writing (or debugging) the initial code (DRIVER) while the other reviews and suggests improvements (NAVIGATOR). **Take turns regularly (every 10 to 15 minutes)** to ensure both of you are actively involved.

PART A - QUESTIONS

- Create a text file named **lab9_a.txt**.
- The first line should have your name.
- Answer the following questions in that text file.

1. The _____ symbol is the dereferencing operator. It is also called the indirection operator.
2. The _____ symbol means “address of” and is called the address operator.
3. The name of an array, without any brackets, acts as a(n) _____ to the starting address of the array.
4. A _____ variable can hold the memory address of a variable.

Given the following information, fill in the blanks of questions 5 through 8 with either “an address” or “3.75”.

```
float * ptr;  
float pay = 3.75;  
ptr = &pay;
```

5. The statement `cout << ptr;` will print _____.
6. The statement `cout << *ptr;` will print _____.
7. The statement `cout << &pay;` will print _____.
8. The statement `cout << pay;` will print _____.
9. If I created another `float` variable like this
`float another_pay = 4.88;`
Can I change the `ptr` variable to point to `another_pay` instead of `pay`? If so, how?
10. If I created an `int` variable like this
`int intPay = 4;`
Can I change the `ptr` variable to point to `intPay` instead of `pay`? If so, how?

PART B – PRACTICE WITH BASIC POINTERS

Write a program named `lab9_b.cpp`. Do the following in the program.

Note that you must use pointers in order to get credit for this program.

- Create five variables.
 - Three integers: length, width, & area.
 - Two pointers: lengthPtr & widthPtr
- Initialize the pointers to hold the address of the integer variables (lengthPtr should point to length and widthPtr should point to width).
- **From this point forward, if you are accessing or changing the data in length or width, you must do that with the pointer variables, lengthPtr & widthPtr.**
- Get user input.
 - Ask the user for the length of the rectangle.
 - Ask the user for the width of the rectangle.
- Calculate the area of the rectangle.
- Print results.
 - Print the area.
 - Print “The length is greater than the width.” If the length of the rectangle is greater than the width.
 - Print “The width is greater than the width.” If the width of the rectangle is greater than the length.
 - Print “The length and width are the same.” If they are the same.

Sample Output

User input is highlighted in **yellow**.

```
Please input the length of the rectangle.
```

```
9
```

```
Please input the width of the rectangle.
```

```
3
```

```
The area is 27
```

```
The length is greater than the width.
```

PART C – VECTOR PRACTICE

You are going to practice with **Vectors** and implementing **algorithms** in this practice assignment.

A **Vector** is not built into the C++ language like an array, but it is similar to an array in that it is a single variable that holds multiple pieces of data of the same data type (a sequence data structure). Also like an array, each piece of data in a vector is called an element.

One algorithm you will use is called the **selection sort algorithm**. There are many sort algorithms in Computer Science. The selection sort algorithm is considered a very simple algorithm (easy to understand, easy to implement) but it is very slow if your list of items that you are sorting is very large.

Write a program named **lab9_c.cpp**. Do the following in the program.

1. Create and initialize a 10-element integer vector to the following values: 7, 4, 36, 1, 3, 6, 2, 99, 77, 10
2. Multiply all elements of the vector by 2.
3. Print out the vector in an easy-to-read format like the sample output.
4. Add up all vector elements.
5. Print out the total.
6. Sort the vector elements using the *pseudocode* for the selection sort algorithm (provided below).
7. Print out the vector again, in the same format as you did earlier.

SELECTION SORT ALGORITHM PSEUDOCODE

The \leftarrow symbol in this pseudocode is assignment.

Pseudocode is not code written in a particular language, but is not written out in English either...it is in-between the two.

Notice this pseudocode is based on the input being an array instead of a vector, but you can modify it for a vector pretty easily.

```
//ALGORITHM: SelectionSort(A[0,...,n-1])
//Purpose: Sorts given array by selection sort
//Input: An array A[] of orderable elements
//Output: An array A[] sorted in ascending order

for i ← 0 to n - 2
    //find the next minimum value in the array
    min ← i
    for j ← i + 1 to n - 1
        if A[j] < A[min]
            min ← j
        end if
    end for
    //swap the current array item with the minimum element
    temp ← A[i]
    A[i] ← A[min]
    A[min] ← temp;
end for
```

SAMPLE OUTPUT

The vector elements, multiplied by 2, are: 14 8 72 2 6 12 4 198 154 20

The total of all array elements is 490

The sorted vector elements are: 2 4 6 8 12 14 20 72 154 198

WHAT TO TURN IN

ZIP & upload the following files to the ilearn submission folder:

- lab9_a.txt
- lab9_b.cpp
- lab9_c.cpp