Lab Work 4
Learning Objectives

- Basics of pointers
- Pointers used with arrays
- Pointers used with functions
- Dynamic Memory
- Smart pointers

Exercise 1: Basics of pointers

1. Explain the output of each code below:
   a) cout << ∑
   b) int *ptr = nullptr;
   c) int sum;
      int *sumPtr = nullptr;
2. Explain what happens in this code

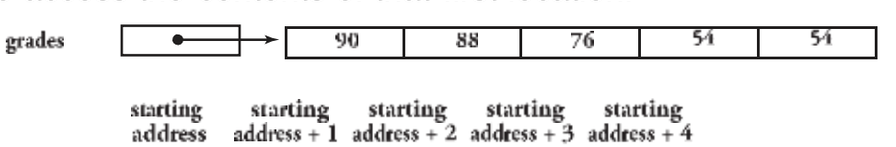
```
void swap(int&first, int&second)
{
int temp;
temp = first;
first = second
}
```

3. Explain what this code does and suggest the output

```
intmain()
{
int one = 10;
int*ptr1 = nullptr;  //  ptr1 is a pointervariable that points to an int

ptr1 = &one;

cout << "The value of one is " << one << endl << endl;cout << "The value of &one
is " << &one << endl << endl;cout << "The value of ptr1 is " << ptr1 << endl << endl;
cout << "The value of *ptr1 is " << *ptr1 << endl << endl;


return 0;
}
```

Arrays and pointers

When arrays are passed to functions they are passed by pointer. An array nameis a pointer to the beginning of the array. Variables can hold just one value and so we can reference that value by just naming the variable. Arrays, however, hold many values. All of these values cannot be referenced just by naming the array. This is where pointers enter the picture. Pointers allow us to access all the array elements. Recall that the array name is really a pointer that holds the address of the first element in the array. By using an array index, we dereference the pointer which gives us the contents of that array location. If grades is an array of 5 integers, as shown below, grades is actually a pointer to the first location in the array, and grades[0] allows us to access the contents of that first location.



Take a look at this program

```
int grades[] = {90, 88, 76, 54, 34};
    // This defines and initializes an int array.
    // Since grades is an array name, it is really a pointer
    // that holds the starting address of the array.

cout << "The first grade is "          // The * before grades
     << *grades << endl;               // dereferences it so that the
                                       // contents of array location 0
                                       // is printed instead of its
                                       // address.

cout << "The second grade is "         //Thesameisdonefortheother
cout << *(grades + 1) << endl;"The third   // elements of the array. In
cout << grade is "                     // eachcase, pointerarithmetic
cout << *(grades + 2) << endl;"The fourth  // gives us the address of the
     << grade is "                     // nextarrayelement. Then the
     << *(grades + 3) << endl;"The fifth   // indirection operator * gives
     << grade is "                     // us the value of what is stored
     << *(grades + 4) << endl;         // at that address.
     <<
```

What is printed by the program?

```
The first grade is 90
The second grade is 88
The third grade is 76
The fourth grade is 54
The fifth grade is 34
```

Exercise

1. The symbol is the dereferencing operator.
2. The ampersand__symbol means “address of.”
3. The name of an array, without any brackets, acts as a(n)____to the starting address of the array.
4. An operator that allocates a dynamic variable is
5. An operator that deallocates a dynamic variable is
6. Parameters that are passed by_____are similar to a pointer variable in that they can contain the address of another variable. They areused as parameters of a procedure (void function) whenever we want a procedure to change the value of the argument.

Given the following information, fill the blanks with either “an address” or “3.75”.

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

7. cout<<pointer; will print an .
8. cout<<*pointer; will print_____.
9. cout<<&pay; will print an
10. cout<<pay; will print

Lab Drills

Drill 1

Create a new cpp file. Add in all the codes below. Try to fill in the blanks

```
// This program demonstrates the use of pointer variables
// It finds the area of a rectangle given length and width
// It prints the length and width in ascending order

// PLACE YOUR NAME HERE

#include <iostream>
using namespace std;

int main()
{

	int length;	// holds length
	int width;	// holds width
	int area;	// holds area

	int *lengthPtr = nullptr;	// int pointer which will be set to point to length
	int *widthPtr = nullptr;	// int pointer which will be set to point to width

	cout << "Please input the length of the rectangle" << endl;
	cin >> length;

	cout << "Please input the width of the rectangle" << endl;
	cin >> width;

	// Fill in code to make lengthPtr point to length (hold its address)

	// Fill in code to make widthPtr point to width (hold its address)

	area =// Fill in code to find the area by using only the pointer variables

	cout << "The area is " << area << endl;

	if (// Fill in the condition length > width by using only the pointer
variables)
		cout << "The length is greater than the width" << endl;

	else if (// Fill in the condition of width > length by using only the pointer
variables)
		cout << "The width is greater than the length" << endl;

	else
		cout << "The width and length are the same" << endl;

	return 0;
}
```