Class Activity 8.2: Pointer Arithmetic and Arrays

In this class activity we will learn about pointer arithmetic and how they relate to arrays.

Consider the code below and the corresponding screen output to answer the questions that follow

	Code	Screen output
1	`#include <iostream>`	
2		
3	`int main() {`	
4	`  int vals[] = {4, 7, 11};`	
5	`  std::cout << vals << "\n";`	`0xfff000bd0`
6	`  std::cout << vals[0] << "\n";`	`4`
7	`  std::cout << *vals << "\n";`	`4`
8	`  std::cout << vals + 1 << "\n";`	`0xfff000bd4`
9	`  std::cout << *(vals + 1) << "\n";`	`7`
10	`  int* ptr_vals = vals; // take note there is no &`	
11	`  std::cout << ptr_vals[0] << "\n";`	`4`
12	`  return 0;`	
13	`}`	

1. Analyze line #5. We see that printing the value of *vals* gives us an address. If a variable stores an address, then what kind of variable is it?

Answer: Pointer

2. Analyze line #6 and 7. What do you notice about the output of both lines? Why do you think dereferencing *vals* (using *) provides that result?

Answer:

The output of both lines is not the address but what is is stored in that index.

It means to get the memory location of that pointer.

3. Analyze line #5 and 8. Why do you think the addresses differ by 4 bytes? *Hint: What is the data type of each element in the array?*

Answer:

Since the data type of each element is an integer, by adding 1 to vals, the address goes up by the first index of the array.

4. Analyze line #9. Why do you think the output is 7? Where does it get this number?

Answer:

Since the the adding of 1 is in parenthesis and overall it is a pointer, the print out will be the next value of the array

5. Analyze line #10. Why do you think it is valid to assign an array to a pointer?

Answer:

An array is an pointer so it would reference a different value within the array

6. Based on everything you've discovered, is there any other way to print out the value 11 in the array using the *ptr_vals* variable?

Answer:

```
cout << *(ptr_vals + 2) << endl;
```

7. How would you display the third element of the array using pointer arithmetic?

```
string the_beatles[4] = {"John", "Paul", "George", "Ringgo"};
```

Answer:

```
cout << *(the_beatles + 2) << endl;
```

8. Create a for loop that goes through a string array and prints it's elements on the screen. Use pointer arithmetic instead of subscripts.

```
string the_beatles[4] = {"John", "Paul", "George", "Ringgo"};
```

Answer:

```
for (int r = 0; r < 4; r++)
    cout << *(the_beatles + r) << endl;
```

9. Consider the following code to answer the questions that follow.

```
int num1 = 10;
int num2 = 10;
int* ptr_num1 = &num1;
int* ptr_num2 = &num2;
```

10. What will the following expression return, True or False? Why?

```
ptr_num1 == ptr_num2
```

Answer:

False, because without they are going to print out different addresses since there is no * in the front.

11. What will the following expression return, True or False? Why?

```
*ptr_num1 == *ptr_num2
```

Answer:

True, because they are both displaying 10 from the address

12. Consider the following variable declarations.

```
double prices1[3] = {100.00, 56.65, 35.43};
double prices2[3] = {100.00, 56.65, 35.43};
```

What will the conditional expression return and why?

```
prices1 == prices2
```

Answer:

False, because an array is a pointer and without an *, it prints out addresses and not the values stored in the address.