Activity #24: Pointers

Recorder's Report

Manager:	Reader:
Recorder:	Driver:
Date:	Score: Satisfactory / Not Satisfactory

Record your team's answers to the key questions (marked with) below.

a) Model 1, Question #6

b) Model 2, Question #10.b

c) Model 2, Question #12.c

Activity #24: Pointers

In this activity, you will work in teams of 3–4 students to learn new concepts. This activity will introduce you to pointers in C++.

Content Learning Objectives

After completing this activity, students should be able to:

- Define what pointers are and explain how they work.
- Explain how to use the reference and dereference operators.

Process Skill Goals

During the activity, students should make progress toward:

- Write C++ code that utilizes pointers.

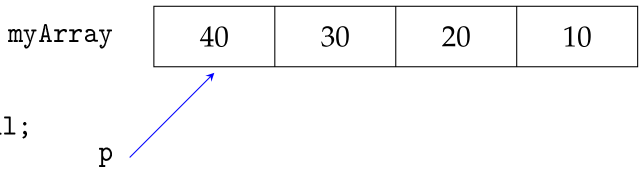
Model 1 A First Program with Pointers

```
#include <iostream>

using namespace std;

int main() {
  int myArray[] = {40, 30, 20, 10};
  int *p = myArray;
  // Your code goes here
  cout << "Value of p: " << p << endl;
  cout << "Value of *p: " << *p << endl;
  cout << "Value of myArray: ";
  for (int i = 0; i < 3; i++) {
    cout << myArray[i] << ",";
  }
  cout << myArray[3] << endl;
}

```



Refer to Model 1 above as your group develops consensus answers to the questions below.

Questions (30 min) Start time:

1. The file `activity24a.cpp` contains this code. Run it and record the output on the lines below.

a) `Value of p:`

b) `Value of *p:`

c) `Value of myArray:`

2. Run the code several times. Describe how (if at all) the output changes each time you run the program.

3. A *pointer* is a variable that holds a memory address, rather than data. The diagram on the right of the model depicts the pointer relationship in this code.

a) Based on the diagram, which variable is a pointer?

b) Now looking at the code, how is a pointer declared in C++?

c) How does this explain what you observed in question 2 above?

d) Does a pointer have a data type like other variables? Explain.

4. Now add each of the following code snippets in line 8, one at a time. Run the code and record its output.

a) `p++`

(a) `Value of p:`

(b) `Value of *p:`

(c) `Value of myArray:`

b) `(*p)++`

(a) `Value of p:`

(b) `Value of *p:`

(c) `Value of myArray:`

c) `*(p+2) += 2;`

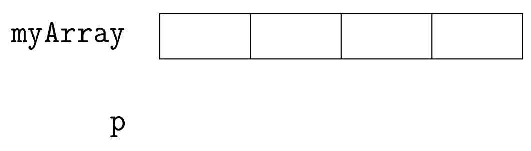
(a) `Value of p:`

(b) `Value of *p:`

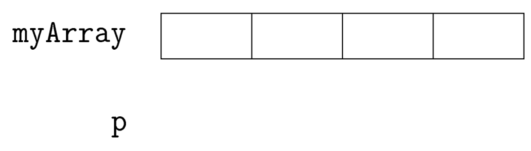
(c) `Value of myArray:`

5. Complete the diagram below to show the contents of `myArray` and the pointer relationship between `p` and `myArray` after line 8 of the model was changed as indicated above.

a) `p++`



b) `(*p)++`



6. Write a single line of code referencing only `p` that does the following when added on line 8.

a) Adds 5 to the first entry in `myArray` (making it 45):

b) Changes the second line of output to "`Value of *p:  10`":

c) Adds 5 to the 2nd entry in `myArray` (making it 35):

7. Finally, place the command `p -= 10;` on line 8 of the model and run the code several times.

a) What happens to the output each time you run the program?

b) This example illustrates one of the *dangers* of using pointers. In your own words, describe what that danger is.

Model 2 A Second Program with Pointers

```cpp
#include <iostream>

using namespace std;

int main() {
  int numOne = 1892;
  int numTwo = 1973;
  int numThree = 2008;
  int *numArray[3] = {&numOne, &numTwo, &numThree};
  int *p;

  p = &numOne;
  cout << "Number: " << *p << endl;
  p = &numTwo;
  cout << "Number: " << *p << endl;
  p = &numThree;
  cout << "Number: " << *p << endl;
}

```

Refer to Model 2 above as your group develops consensus answers to the questions below.

Questions (20 min) Start time:

8. Without running any code, predict what the output of this program will be.

9. The code for this model can be found in the file `activity24b.cpp`. Run this code and see if your predictions are correct.

10. When dealing with pointers in C++, the symbol `*` is called the *dereference operator*. When it is placed in front of a pointer variable, it retrieves the data to which the pointer variable points.

a) On which lines in the model above is the dereference operator used?

b) How would the output of this program change if the dereference operator were removed from those lines?

c) How does C++ tell the difference between a `*` that is a dereference operator and a `*` that indicates multiplication (i.e. `int x = 2 * y;`?

11. The symbol `&` is called a *reference operator*.

a) On which lines in the model above is the reference operator used?

b) In your own words, describe what the reference operator does.

c) Is this usage of symbol `&` related to its usage to indicate “pass-by-reference” parameters in a function? Explain.

d) How would things changed if the reference operators were removed from lines 12, 14, and 16 of the model?

12. Consider the array `numArray` defined in this model.

a) What is the type of this array?

b) Describe the elements that are stored in this array.

c) Sketch a pointer diagram, similar to that seen in model 1, that depicts the values and pointer relationships of all five variables defined in model 2.

13. Suppose a new variable was defined as follows: `int** p2 = &numArray[1]`.

a) Describe both the type and purpose of the variable `p2`.

b) What would be the output of each of the following commands?

   (a) `cout << p2 << endl;`

   (b) `cout << *p2 << endl;`

   (c) `cout << **p2 << endl;`

c) Sketch a pointer diagram depicting the relationship between the variable `p2` and the rest of the variables in this model.