Lecture 10: Pointers

Workflow 01

For each step, make sure that the program runs before continuing.

Replit Project: Pointers

Setup

1. Before you begin, make sure to include the iostream, string, ctime and cstdlib libraries, and then, write the

statement

using namespace std;

below them.

Pointer Declaration

2. In the main function, declare two int variables, two int arrays and four int pointers.

3. Afterwards, for each declaration, write a cout statement in the format below

cout << ''Address of object : ''<< &object << '\n';

where object is the name of a variable, array or pointer that was declared in step 2. Likewise, all object

nonterminals in the syntax must be the same value. For instance, if one of the variables were named value1,

then the cout statement should be

cout << ''Address of value1: ''<< &value1 << '\n';

Pointer Assignment

4. Write the statement

srand(time(NULL));

at the beginning of the body of the main function.

5. Before the main function, write the function

int RandomAssign()

{

return ((rand() % 100) + 1);

}

6. Between the RandomAssign() function and the main function, define the function RandomAssignments()

whose header is

void RandomAssignments(int data[],int n)

Given that n represents the size of data, the function assigns a random value between 1 and 100 inclusively to

each element of data by assigning each element RandomAssign() caller.

7. Assign random values to the variables and arrays that were declared.

8. Assign the variables and arrays to the pointers so that no pointer is referencing the same thing.

9. For each pointer, write a cout statement in the format below

cout << pointer << '': ''<< \*pointer << '\n';

10. Using the pointers referencing the variables, change the values of the variable to twice their current values,

and then, for each variable, write a cout statement in the format below

cout << &variable << '': ''<< variable << '\n';

11. Assign a pointer to another pointer, and then, write the cout statement from step 9 for the pointer that was

assigned a new value.

Pointer Initialization

12. Initialize three new pointers. One must be initialized to a variable, another to an array, and the last to a

pointer.

13. For each of the new pointer variable, write a cout statement in the format below

cout << ''pointer : ''<< pointer << '\n';

where both pointer are the same pointer identifier. For instance, if one of the new pointer is named nn, you

would write

cout << ''nn: ''<< nn << '\n';

Constant Pointer

14. Declare three new pointers. They should be

• An int readonly pointer. Assign it a value separately.

• An int constant pointer.

• An int readonly constant pointer.

15. Repeat step 13 for each of these new pointers.

Dynamic Memory

16. Assign one of the nonconstant pointer a dynamic variable and another nonconstant pointer a dynamic array.

17. Call RandomAssign() to assign a value to the dynamic variable.

18. Call RandomAssignments() to assign values to the elements of the dynamic array.

19. Write the cout statement

cout << pointer << '': ''<< \*pointer << '\n';

where pointer is the dynamic variable pointer.

20. Write the statements

cout << pointer << '':\n'';

Print(pointer );

where pointer is the dynamic array pointer.

21. Deallocate both the dynamic variable and dynamic array pointers, and then, make them null pointers.

Paasing By Address

22. Before the main function, define the void function Swap() whose header is

void Swap(int \*a,int \*b)

It swaps the values of the variables pointed to by the parameters.

23. In the main function, call Swap() with the two variables as the arguments. Before and after the call, display

their values.