**Pseudo Code**

main() {

int<1-3>

pInt<1-3>

// Get user inputs and store to ints.

Int<1-3> = get\_user\_input();

// Reference ints holding values.

\*pInt<1-3> = int<1-3>;

Print pInt<1-3>

// Free memory for pointers.

delete(pInt<1-3>);

exit;

}

int get\_user\_input()

{

int ret;

ret = input if input is an int;

return ret;

}

**Src Code:**

/\*\*

\* AUTHOR: Alec McDaugale

\* DATE: 10-26-2014

\*

\* DESCRIPTION:

\* This program asks the user to input three integer values. The values are

\* stored in three different variables. Each variable has an integer pointer to

\* dynamic memory. The contents of the variables and thei pointers are then

\* displayed to the user.

\*

\* NOTE:

\* The new operator and delete operator need to be used for management.

\*/

// Max inputs.

#define INPUT\_COUNT 3

#include <iostream>

#include <cstring>

using namespace std;

int get\_user\_input(int intNum);

/\*\*

\* Main function.

\*/

int main() {

// Variable declarations.

int int1;

int int2;

int int3;

int \*pInt1 = new (nothrow) int;

int \*pInt2 = new (nothrow) int;

int \*pInt3 = new (nothrow) int;

// Get user inputs and store to ints.

int1 = get\_user\_input(1);

int2 = get\_user\_input(2);

int3 = get\_user\_input(3);

// Reference ints holding values.

\*pInt1 = int1;

\*pInt2 = int2;

\*pInt3 = int3;

// Print results.

cout << "Address pointed to by pInt1: " << pInt1 << endl;

cout << "Value pointed to by pInt1: " << \*pInt1 << endl;

cout << "Address pointed to by pInt2: " << pInt2 << endl;

cout << "Value pointed to by pInt2: " << \*pInt2 << endl;

cout << "Address pointed to by pInt3: " << pInt3 << endl;

cout << "Value pointed to by pInt3: " << \*pInt3 << endl;

// Free memory for pointers.

delete(pInt1);

delete(pInt2);

delete(pInt3);

// Return success.

return 0;

}

/\*\*

\* Gets the user input and garuntees it is of type int.

\*

\* param: intNum - The number int to prompt for.

\*

\* return: int - The int input.

\*/

int get\_user\_input(int intNum)

{

int ret = 0;

cout << "Please enter " << intNum << " int." << endl;

cin >> ret;

// Sanitize input to garuntee input is an int.

while ( cin.fail() )

{

cout << "Please enter a numeric value." << endl;

cin.clear();

cin.ignore(256, '\n');

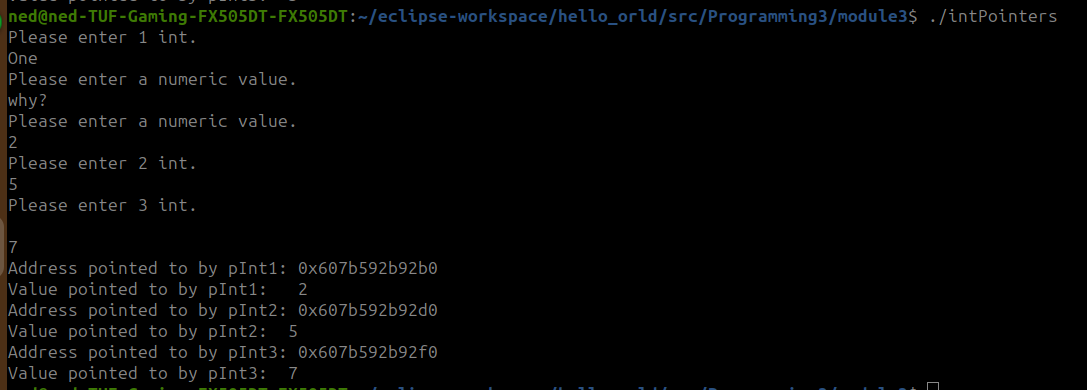
cin >> ret;

}

return ret;

}

**Execution Screenshot:**



**Github:**

<https://github.com/amcdaugale/Programming3/blob/main/module3/intPointers.cpp>

