Module 3 Critical Thinking Assignment

Programming III

CSC450-1 Fall 2025 Session C

Dr. Bindu George

Grant Brosovich

09/28/2025

**IntPointer:**

//============================================================================

// Name : IntPointer.cpp

// Author : Grant Brosovich

// Version : 1

// Copyright : Some kind of Copyright

// Description : Integer input with pointers

//============================================================================

/\*

\* I used functions to accomplish the pointer assignments from input variable in the main function.

\* I also made a function to incorporate error handling and allow for the user to continue entering values until an integer is input.

\* The error handling function also does a loop until a valid input is found.

\* lastly, a catch-all in the main when the dynamicAssignDisplay() function is called for any other errors.

\*/

**#include** <limits>

**#include** <stdexcept>

**#include** <iostream>

**using** **namespace** std;

**void** **dynamicAssignDisplay**(**int** varA, **int** varB, **int** varC) {

//This section assigns a pointer to a new value using the input variables.

**int**\* pointA = **new** **int**(varA);

**int**\* pointB = **new** **int**(varB);

**int**\* pointC = **new** **int**(varC);

//This is the output section that outputs both the input variable var# and the pointer value point#

cout << "First input value: " << varA << " | pointer value: " << \*pointA << **endl**;

cout << "Second input value: " << varB << " | pointer value: " << \*pointB << **endl**;

cout << "Third input value: " << varC << " | pointer value: " << \*pointC << **endl**;

//Memory management section

**delete** pointA;

**delete** pointB;

**delete** pointC; }

**int** **getValidInput**(**const** string& message) {

string input;

**int** value;

**while** (**true**) {

cout << message;

**getline**(cin, input);

**if** (input == "q" || input == "exit") {

cout << "Exiting.";

**exit**(0);

}

**try** {

//This identifies the size of the characters and if there are characters other than digits in the entry

size\_t pos;

value = stoi(input, &pos);

**if** (pos != input.length()) {

**throw** invalid\_argument("More characters found after numbers.");

}

//Error handling for non-integer entries other than q and exit

//Also catches too large entries

**return** value;

} **catch** (invalid\_argument&) {

cout << "Invalid input. Please enter a number or q to quit. \n";

} **catch** (out\_of\_range&) {

cout << "Number out of range. Please enter a number or q to quit. \n";

}

}

}

**int** **main**() {

**int** one, two, three;

cout << "Enter three integers please or enter q any time to quit." << **endl**;

one = getValidInput("First integer: ");

two = getValidInput("Second integer: ");

three = getValidInput("Third integer: ");

**try** {

dynamicAssignDisplay(one, two, three);

} **catch** (...) {

cerr << "An unexpected error occurred while processing the integers." << **endl**;

**return** 1;

}

**return** 0;

}

**Psuedocode:**

Start IntPointer Program

-Declare assignment function

  1) build pointers for each input variable

  2) display values for the input variables and the pointer variables

  3) delete pointer variables

-Declare input validation function

  1) declare local variables: input, value

  2) start loop

  1: display initial message

  2: look for exit entry

  exit

  3: check to see if non-integer values have been entered

  display error message and return to loop beginning if non-integer characters found

  return value variable if only integer variables found

  4: check to see if the integers are too many and may cause overflow errors

  display error message and return to loop beginning

-Declare Main function

  1) declare variables: one, two, three

  2) display input request message

  3) assign variables using input validation function

  4) check to see if using variable in assignment function causes errors

  1: if errors present, exit with failure code

  2: if none, exit with success code

A screenshot of a computer program

AI-generated content may be incorrect.

User input to pointers with memory management through delete() and error handling at multiple stages. Ran as C++ application that compiled and ran successfully.