CS 150 Lecture 17 Exercises

Complete each of the exercises below and upload to Canvas before the deadline.

# A. Writing Functions with vector

# Open **vectors.h** where you'll find the prototypes and documentation for three **vector** functions. Implement the functions in **vectors.cpp** and then run the tests. (You can put your own code inside the **run()** function at the bottom of **vectors.cpp**.) When you're finished, paste the requested information in the text areas below.

|  |
| --- |
| *Copy and paste a screenshot of source code of vectors.cpp* |

|  |
| --- |
| *Copy and paste a screenshot of the test results* |

B. Multi-file Memory Layout and Management

Open **f1.cpp** and **f2.cpp**. Use **make** **run** to compile, link and run the program. Use the files to **answer the following questions**. Use this as notes to yourself to study later. The goal is to identify the **scope**, **duration** and **linkage** for each item used in the program and understand exactly what this means. (Reverse each change after examining it.)

In **f2.cpp**, remove the word **extern** from line 1. What is the error and why does it occur?

|  |
| --- |
| Answer here  multiple definition of `a'. Only one definition of each object in an executable. One Definition Rule or the ODR. |

Remove the word **static** from the definition of **b** on line **5** in **f1.cpp**. Does the program still compile and link? Does the program output run differently. Does the change to **b** on **f2.cpp:9** have any effect, since **b** now has **external** linkage?

|  |
| --- |
| Answer here  No difference. Runs the same. |

Can you remove the word **static** from both definitions of **b**? What happens. Why?

|  |
| --- |
| Answer here  Error. multiple definition of `b'; breaking ODR. |

Add a declaration for **d()** to **f2.cpp** and call the function from inside **main()**. What happens? Why?

|  |
| --- |
| Answer here  undefined reference to `d()'. Function has internal linkage. Can only be used in file f1.cpp |

Can you add an **extern** declaration to **f2.cpp** for the variables **e** and **f** declared inside **f1.cpp**? What happens if you change those variables in **f2.cpp**? (Show me.)

|  |
| --- |
| Answer here  /workspaces/cs150-su23-anguyen798/cs150/ic/LEC-17/B/f2.cpp:8: undefined reference to `e'  /usr/bin/ld: /workspaces/cs150-su23-anguyen798/cs150/ic/LEC-17/B/f2.cpp:9: undefined reference to `f'  Both variables are locals, so they have no linkage. Can’t be used outside of function d(). |

Does the program produce different output (run differently) if you remove the **static** from the definition of the local variable **f**? Why?

|  |
| --- |
| Answer here  F no longer retains its value between function calls because it now has automatic storage instad of static storage. |

What happens if you move the definition of **c()** above the definition of **d()**? Show me.

|  |
| --- |
| Answer here  f1.cpp:9:5: error: ‘d’ was not declared in this scope  9 | d();  | ^  Using d before defined or declared so it is out of scope. |

C. Variable Sizes & Pointers

Open **variables.cpp** where you'll find a program that creates several variables. Print the address and size of each variable and the value of each pointer, as directed in class. Copy your completed code into the text areas.

|  |
| --- |
| Copy and paste a screenshot of source code for variables.cpp |

|  |
| --- |
| Copy and paste a screenshot the main function running (testing) |

D. Pointers and Functions

Open **pointers.cpp**. Follow the instructions to write a function to use a pointer for output. When you are finished, show me your code and the program running. Copy the results into the text area below.

|  |
| --- |
| Copy and paste a screenshot of source code of pointers.cpp |

|  |
| --- |
| Copy and paste a screenshot the program running |