# Lab 01 Quiz

Started: Sep 8 at 10:23pm

# Quiz Instructions

Download and complete the Lab 01 assignment (found in **CANVAS -> Files -> Lab**) with your group, and then submit the answers to the multiple choice questions here.

## Welcome to 281 - Getting to Know You!

Before beginning this week's quiz, please fill out this short survey: **https://forms.gle/yTrHUDsTVtSvTjs17 (https://forms.gle/yTrHUDsTVtSvTjs17)**

This survey should not take more than a few minutes.

## Section 1 - Logistics

Some questions to ensure that you are aware of important dates and course policies.

Question 1 0.25 pts

What is the due date of project 1?

○ 9/15/2024

● 9/17/2024

○ 9/19/2024

○ 10/10/2024

Question 2 0.25 pts

What is the date of the midterm exam?

○ 10/10/2024

○ 10/15/2024

●

10/17/2024

○

10/22/2024

Question 3 0.25 pts

What percentage of your grade is each project worth?

○

1

○

5

●

10

○

20

## Section 2 - Tools

How to use the tools in EECS 281, including Makefiles, valgrind, and performance profiling tools.

Question 4 0.25 pts

What is the correct login string for CAEN?

●

ssh <uniqname>@login.engin.umich.edu

○

ssh <uniqname>@engin.umich.edu

○

ssh <uniqname>@caen.umich.edu

○

ssh <uniqname>@umich.edu

Question 5 0.25 pts

Why do we use Makefiles?

○

They can create the submission files automatically

○

They automate the (sometimes long and complex) compilation process

○

They can run custom testing scripts

●

All of the above

Question 6 0.25 pts

Will your program compile on the autograder without the project identifier?

○ Yes

● NO, and I'll waste the submission

Question 7 0.25 pts

What is the Makefile command to create a submission file that includes custom test cases?

○ make

○ make all

○ make partialsubmit

● make fullsubmit

Question 8 0.25 pts

Which of the following is a debugging tool?

○ Makefile

○ perf

● valgrind

○ cin

Question 9 0.25 pts

What does perf do?

○ Detects segmentation faults in a program and siplays line numbers that they occurred at.

● Profiles program execution time.

○ Profiles program memory usage.

○ Automates the testing process.

## Question 2.10

Follow the steps outlined in the lab assignment, then answer the following questions.

### Question 10 1.5 pts

What is the **FIRST** error message reported by `valgrind`?

○

`Invalid read of size 4`

○

`Invalid write of size 4`

○

`Use of uninitialised value of size 8`

●

`Conditional jump or move depends on uninitialised value(s)`

○

(no errors are reported, the program runs fine and outputs ten 1-digit integers)

### Question 11 1.5 pts

Which line of code (in `lab1_bad.cpp`) does `valgrind` report the first error occurring on?

○

Line 60

○

Line 65

○

Line 73

○

Line 76

●

Line 79

# Section 3: Debugging (1.35 points)

We have found in previous semesters that students entering 281 are not comfortable in their development environments, specifically with using git, gcc/clang, make, valgrind and debuggers. These tools and more are vital for success in projects and labs in 281, but also in future EECS courses you may take, and in industry. This assignment uses small toy programs to help you get more comfortable and hopefully learn a thing or two before jumping into more complex projects. For each question, assume that the given code is compiled according to the flags given in the Makefile (shown below), under the CAEN Linux environment. **Assume that all the programs are complete and that no other source files exist.**

Makefile flags: `CXXFLAGS := -std=c++1z -Werror -Wall -Wconversion -Wpedantic`

**Feel free to try compiling, running, debugging, and valgrind-ing the code.** Google and StackOverflow are your friends for any errors you find as you answer these questions, so we will require that you use these resources and try to figure out the answers on your own before asking for help in office hours or Piazza. Don't just blindly copy and paste errors into your search engine though, try to really understand what they are trying to tell you. Good luck!

Each question in this section is worth **0.15 points**. Select the best answer (there is only 1 correct option per question).

Not all of the snippets may be buggy!

Question 12 0.25 pts

What is wrong with the following program?

```
int add(int a, int b);
int main() {
   int x = 1;
   int y = 5;
   return add(x, y);
} // main()
```

●

`add()` has no definition

○

`add()` is called with `x` and `y`, but accepts `a` and `b`

○

the program does not compile because `main()` cannot return a value of 6

○

a function cannot be called after a `return` statement

○

nothing is wrong with this program

Question 13 0.25 pts

What is wrong with the following program?

```
int main() {
   int x = 0;
   return x + y;
} // main()
```

○ `main()` cannot return the result of an arithmetic expression

○ `main()` cannot return `x + y`, since `x + y` is a `double`

● `main()` doesn’t contain a declaration for `y`

○ `main()` cannot return an integer

○ nothing is wrong with this program

## Question 14 0.25 pts

What is wrong with the following program?

```
#include <vector>

int *get_some_ints() {
   std::vector<int> ints = {1, 2, 3, 4, 5};
   return ints.data();
} // get_some_ints()

int main() {
   int *some_ints = get_some_ints();
   delete[] some_ints;
   return 0;
} // main()
```

● the memory pointed to by `some_ints` is freed twice

○ `main()` leaks the memory pointed to by `some_ints`

○ a function cannot return a pointer

○ `some_ints` is a pointer and not an array, so `delete` should be used instead of `delete[]`

○ nothing is wrong with this program

## Question 15 0.25 pts

What is wrong with the following program?

```
struct Thing {};
int main() {
   Thing a;
   Thing b;
   bool less = a < b;
} // main()
```

○ main() cannot declare an instance of a Thing object

○ main() must have a return statement

○ Thing cannot have an empty definition, so this code does not compile

● main() tries to use the < operator, which is not defined for the Thing type

○ nothing is wrong with this program

## Question 16 0.25 pts

What is wrong with the following program?

```
int main() {
   int some_ints[5] = {1, 2, 3, 4, 0};
   for (int *p = some_ints; p; ++p)
      *p = 0;
   return 0;
} // main()
```

● main() indexes out of bounds into some_ints

○ main() leaks the memory pointed to by some_ints

○ some_ints points to uninitialized memory

○ the for loop is missing a curly brace, so this code will not compile

○ nothing is wrong with this program

## Question 17 0.25 pts

What is wrong with the following program?

```
#include <iostream>

int sum_ints() {
   int *some_ints = new int[50];
   for (int i = 0; i < 50; ++i) {
      some_ints[i] = i;
   } // for i
   int sum = 0;
   for (int i = 0; i < 50; ++i) {
      sum += some_ints[i];
   } // for i
   return sum;
} // sum_ints()
```

```
int main() {
   std::cout << sum_ints();
} // main()
```

○ `sum_ints()` indexes out of bounds into `some_ints`

● `sum_ints()` leaks the memory pointed to by `some_ints`

○ `some_ints` points to uninitialized memory

○ `main()` must have a `return` statement

○ nothing is wrong with this program

Question 18 0.25 pts

What is wrong with the following program?

```
int factorial(int n) {
   return n * factorial(n - 1);
} // factorial()

int main() {
   return factorial(3) - 6;
} // main()
```

○ `factorial(3)` returns an uninitialized value

○ a mathematical operation cannot follow a `return` statement

● `factorial(3)` never returns (and may cause a stack overflow)

○ `main()` cannot return the result of a recursive function

○ nothing is wrong with this program

Question 19 0.25 pts

What is wrong with the following program?

```
#include <iostream>

int what_is_2x281() {
   int x, y = 281;
   return x += y;
} // what_is_2x281()

int main() {
    std::cout << "What is 2 x 281?\n" << what_is_2x281();
} // main()
```

○

the += operator cannot be used after a `return` statement, since x would be returned *before* being added to y

●

`what_is_2x281()` returns an uninitialized value

○

the type of y is undefined

○

`main()` must have a `return` statement

○

nothing is wrong with this program

Question 20 0.25 pts

What is wrong with the following program?

```
#include <iostream>

void takes_an_integer(int x) { }

int main() {
    size_t x;
    std::cin >> x;
    takes_an_integer(x);
} // main()
```

○

The code does not compile because `takes_an_integer()` has an empty definition.

○

`main()` must have a `return` statement

●

`print_an_integer()` takes an `int` but is called with a `size_t`, which may cause loss of precision

○

`size_t` is not a valid type

○

nothing is wrong with this program

## Section 4 - File Input & Output

Refer to the assignment regarding the code for the next two questions.

Question 21 0.25 pts

Which line(s) of code would read an integer from a file using file redirection?

○ `ifstream readfile; readfile >> i;`

○ `cout << i;`

● `cin >> i;`

○ `ofstream writefile; writefile << i;`

Question 22 0.25 pts

Which of the following command line commands would run the above program (in main.cpp, already compiled to the executable main) with input file redirection?

○ `make all`

○ `./main input_file.txt`

○ `./main > input_file.txt`

● `./main < input_file.txt`

Question 23 0.25 pts

What will be the difference, if any, between the two resulting `string` variables?

○ `S2` will have a trailing space, while `S1` will not.

● Both strings will be exactly the same.

○ `S1` will have a newline character at the end, while `S2` will not.

○ `T1` will not read the file.

Question 24 0.25 pts

What if there is a space at the end of each word in the file?

● `S2` will contain a trailing space, while `S1` will not.

○ Both strings will be exactly the same.

○ `S1` will have a newline character at the end, while `S2` will not.

○ T1 will not read the file.

## Section 5 - Getopt Long

Answer the following three questions on Getopt below. If you are unsure about this topic, please review the SundaySundaySunday lecture recording. Each question is worth **0.25 points**.

Question 25 0.75 pts

You're writing a simple text-based game on the command line. When the user runs your executable (`./281quest`) the game starts at the first level. If the user wants to skip to a certain level, they can instead run the program with the command `./281quest --level 5` (where '`--level`' is followed by the level they want to skip to).

Which of the following correctly specifies this flag for getopt?

○ `{"level", no_argument, nullptr, 'l'}`

○ `{"level", optional_argument, nullptr, 'l'}`

● `{"level", required_argument, nullptr, 'l'}`

Question 26 0.75 pts

You're writing a program that will take five command line options: **-p** --paoletti, **-d** --darden, **-a** --angstadt, **-m** --markov, and **-g** --garcia. Options **d**, **m**, and **g** will have required arguments. The rest take no arguments. Which of the following strings is a correct short options string?

○ pd:am:g

○ p:d:a:m:g

○ p:dam:g:

● pd:am:g:

○ p:da:mg

## Section 6 - Handwritten Question (5 points)

Complete the handwritten problem detailed in the lab assignment document and turn it in before the end of lab.  We recommend you complete the problem on your own and then discuss your solution with your groupmates!

## Section 7 - Coding Assignment - Music Sorting (5 points)

For this lab, there will be a coding portion that is worth 10 points. Please follow the instructions in the lab 1 assignment sheet for instructions on how to complete this portion. Submit your solution to the autograder.

If you have completed questions 1 to 18, you are done with the "Canvas quiz" portion of this lab assignment. Check your answers before submitting. You will only have **three** attempts on this quiz (or five if you are enrolled in EECS 403), and your highest score will be kept.

Quiz saved at 10:26pm Submit Quiz