

- 
- Part A consists multiple-choice questions, worth 1 point per question, where at least one answer option is correct. If you answer incorrectly or do not select the exact number of correct options, you will receive zero points for that question.
  - Part B consists of questions with varying point values.
  - Grade boundaries: E: 10, D: 12, C: 14, B: 16, A: 18, out of a maximum of 20.
- 

## Part A: multiple choice

*Please gather the answers for Part A on a single answer sheet.*

1. What do programmers mean with the word “stderr”?
  - A. It is a reference to informal programming standards.
  - B. It is a kind of file to write errors and warnings to.
  - C. It is a common programming mistake.
  - D. It is the estimated number of bugs in a large programming project.
2. What does “linking” mean in the compilation process?
  - A. It ensures that a .cpp file is compiled by a C++ compiler.
  - B. It ensures that pointers are handled correctly when allocating dynamic memory.
  - C. It ensures that all .cpp files are compiled.
  - D. It creates a final executable file from several object files.
3. What does “redirection” mean in Unix?
  - A. Output is saved in a named file.
  - B. Output from one program is moved to another program, without saving in a file.
  - C. Input is taken from a different directory than the code resides in.
  - D. An input file is empty, so a program reads from the keyboard instead.
4. What is the programming term for when a function calls itself?
  - A. Segmentation fault
  - B. Returning values
  - C. Recursion
  - D. Exception
5. What is a “Makefile”?
  - A. A file with text explanations of data, separate from the data itself.
  - B. A file describing how a C++ project is compiled.
  - C. A file containing C++ declarations.
  - D. A file that is executable.

## Part B: general questions

Please use a separate sheet of paper for each question in Part B.

6. You are given a large data file `starfield.csv` containing lines of numbers, with each line representing an observation. Write Unix commands to solve the following questions.
- A. Show how to count the number of lines in the data file. (1p)
  - B. Show how to determine whether the number "1.2345" appears in the data file. (1p)
  - C. Show how to sort the observations, by the first number on each line, and outputting the ten lines with lowest first number. (2p)
7. Describe the difference between an interpreted and a compiled language. (1p)
8. Write the C++ program `hello.cpp` that greets each name given as arguments to the program on the command-line. If no name is given, an instruction for how to use the program should be given. (3p)

After compiling to the executable `hello`, it should work like this:

```
$ ./hello
Usage: hello <name>+
$ ./hello Ali Bo Cia
Hello Ali!
Hello Bo!
Hello Cia!
```

9. Write the class `Interval` for representing intervals on the real line. It should have two `float` attributes `left` and `right`, that are set in the constructor, which takes the two end points as parameters.

There should also be a method `intersecting(Interval& other)` that returns `true` if the other interval is intersecting with the interval it is compared with, and `false` otherwise. (4p)

Ensure that the code below would work with your class.

```
int main() {
    Interval unit_interval = Interval(0.0, 1.0);
    Interval some_interval = Interval(0.5, 0.6);
    Interval more_interval = Interval(0.8, 1.2);

    if (some_interval.intersecting(unit_interval)) {
        cout << "(0.5, 0.6) intersects the unit interval." << endl;
    } else {
        cout << "No intersection of (0.5, 0.6) and (0.0, 1.0)" << endl;
    }

    if (more_interval.intersecting(some_interval)) {
        cout << "(0.8, 1.2) intersects (0.5, 0.6)" << endl;
    } else {
        cout << "(0.8, 1.2) does not intersect (0.5, 0.6)" << endl;
    }
}
```

and, if compiled to the executable `interval`, the output is:

```
$ ./interval
(0.5, 0.6) intersects the unit interval.
(0.8, 1.2) does not intersect (0.5, 0.6)
```

10. The C++ code below computes the  $n$  first Fibonacci numbers, where  $n$  is given on the command line. However, there are three mistakes in the code: describe them. (3p)

A reminder: The Fibonacci series starts as 0, 1, 1, 2, 3, 5, 8, ..., and the  $i$ th number is the sum of the two previous Fibonacci numbers.

```
#include <iostream>
using namespace std;

vector<int> fibs(int num) {
    vector<int> fib_vec(num);
    int a = 0;
    int b = 1;

    for (int i=0; i<num; i++) {
        fib_vec[i] = a;
        a = b;
        b = a + b;
    }

    return fib_vec;
}

int main(int argc, char* argv[]) {
    int num;
    if (argc == 0) {
        num = argv[1];
    } else {
        cerr << "Error, no arguments to program" << endl;
        exit(1);
    }

    vector<int> myfibs = fibs(num);
    for (auto num : myfibs) {
        cout << num << " ";
    }
    cout << endl;
}
```