MATEMATISKA INSTITUTIONEN
STOCKHOLMS UNIVERSITET
Avd. Beräkningsmatematik
Kursledare: Lars Arvestad
Examinator: Lars Arvestad

Tentamen i
DA4007 Programming techniques II
7.5 hp
2025-10-23

- 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.
Do not mark your choices on *this* paper! It will be discarded by the exam administrators!

1. What is *generic programming*?

A. Functions and classes get types as parameters.
B. You write code that follow Unix conventions.
C. The code for input is not dependent on the file format.
D. You are making good choices for your program structure, for example using a good class hierarchy.

2. Which of the following is a valid reason to use version control?

A. To speed up compilation of programs.
B. To manage collaborative development and track changes.
C. To protect source code access, typically using encryption.
D. To automatically generate documentation.

3. What does the Unix command `cat` do?

A. It removes duplicate lines.
B. It sorts lines of text alphabetically or numerically.
C. It counts the number of lines in a file.
D. It reads and outputs file contents.

4. In C++, what is the purpose of a constructor?

A. To destroy an object.
B. To initialize an object when it is created.
C. To overload operators.
D. To handle exceptions.

5. What determines whether commands in a Makefile are actually run?

A. The dependencies have been updated more recently than the target.
B. A target file gives a segmentation fault.
C. There is an environment variable specifying which target to create.
D. All commands always run when starting `make`.

Part B: general questions

Please use a separate sheet of paper for each question in Part B.
Do not worry about details like including the right header files or what namespace is being used. You are not required to document your code, but it is in your interest that the grading teacher can understand your solution. This can be achieved with code comments and/or remarks before or after the code.

6. You are given a file `data.txt` with one word per line. Write Unix commands to:

A. Count how many lines contain the word "error". (1p)
B. Copy the the last 10 lines of the file into the new file "last10.txt". (1p)
C. Sort the lines and remove duplicates. (1p)

7. Write a C++ program `palindrome.cpp` that checks if the first command-line argument is a palindrome. Remember that a palindrome is a string that is identical when reversed. (3p)

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

```
$ ./palindrome radar
Yes
$ ./palindrome hello
No
$ ./palindrome
Usage: palindrome <string>
```

8. Write a class `Rectangle` in C++ with attributes `width` and `height`, and a method `area()` that returns the area. Ensure the following code works: (3p)

```
int main() {
    Rectangle r(4.0, 5.0);
    cout << "Area: " << r.area() << endl;
}
```

The output of this example should of course be 20.

9. The following C++ function is meant to count vowels in a string, but there are mistakes. Identify and explain two mistakes. English is assumed as the input language. (2p)

```
string count_vowels(string s) {
    int count = 0;
    for (int i = 0; i <= s.size(); i++) {
        if (s[i] == 'a' || s[i] == 'e' || s[i] == 'i' ||
            s[i] == 'o' || s[i] == 'u') {
        count++;
        }
    }
    return count;
}
```

10. In this problem you are supposed to write a Unix tool for sorting numbers.

A. Write a program `sortnums.cpp` that reads floating point numbers from stdin and prints them in sorted order, from smallest to largest. (3p)

Example:

```
$ ./sortnums < numbers.txt
-1.0
1.5
3.1
```

In this example, `numbers.txt` contains three lines with the numbers 3.1, -1.0, and 1.5, in that order.

B. Modify the program to handle invalid input using exception handling. If a line cannot be converted to a float, print "Invalid input" to stderr and exit. (1p)