

# CS 246 Spring 2017 Midterm Review

June 18, 2017

These are some questions to help you prepare for the midterm. This review material does not cover every single topic; it's your responsibility to ensure that you are prepared for the midterm.

## 1 Bash and Regular Expressions

1. Write the regular expression that would match the following:

- Lines that contain possible course codes of a second year level computer science courses (CS 2XX where XX are digits) at the University of Waterloo;
- Lines that contain possible course codes of computer science courses at the University of Waterloo that contains at least two 4's;
- Lines that contain only a whitespace-delimited list of integers;
- Recall the string representation of a polynomial in A3Q3:

`-? +<term>( +<operator> +<term>)*`

Replace `<term>` and `<operator>` with regular expressions matching these parts so that this "pseudo-regular-expression" becomes an actual regular expression matching such string representation.

2. Given "x=ls", what will the following lines produce when typed in the command line shell? Write ERROR if it produces an error. Assume the current directory contains three files: a, b, c.

- (a) `ls "*"`
- (b) `$(x)`
- (c) `$x`
- (d) `x`
- (e) `echo $($x)`

## 2 Bash Programming Question: permissiveRunSuite

Write a script `permissiveRunSuite` that is invoked as follows:

```
./permissiveRunsuite suite-file program
```

The script is similar to the `runSuite` you wrote in Assignment 1 with the difference being that if an input (`.in` file) is missing or unreadable, it will run the program with no input being redirected. If an output (`.out` file) is missing or unreadable, it considers the expected output to be empty. You may assume that the program ignores all command-line arguments.

A sample run of `permissiveRunSuite` will read the whitespace-separated stems in `suite-file`, and for each stem (say `test`) it will try to find the corresponding `.in` and `.out` files (in our example, `test.in`, `test.out`). If one of the files is not found or not readable, it will treat as the file exists but is empty. Then the script should run the program with the appropriate input and store the program output in some temporary file. The content of the temporary file is compared with the expected output.

If the output of a given test case differs from the expected output, print the following to standard output (assuming the test `test2` failed):

```
Test failed: test2
Input:
(contents of program input, or just print a newline if no input)
Expected:
(contents of actual output, or just print a newline if no output)
Actual:
(contents of the actual program input)
```

with the (contents ...) lines replaced with actual file contents, as described.

You should print an appropriate usage message and exit with an appropriate status code in case the script is not executed with the correct number of arguments. You may create any number of temporary files in any location, as long as you delete all the temporary files created.

*Hint: you may create empty temporary files using `mktemp` with no arguments. This command will print the name of the file to standard output.*

## 3 Programming Question: User ID Generator

Write a User ID generator that will convert names from the following format:

```
<first_name1> <first_name2> ... <first_namen> <last_name>
```

where each name follows the following regular expression: `[A-Z][a-z]*` to the following:

```
<last_name>, <first_name1> <first_name2> ... <first_namen> (<userid>)
```

Where <userid> follows the following format:

- take the first letters of the first names of the person, in lower case;
- append the full last name with first letter in lower case.

Each name takes one line. You may assume that it's allowed for different users to have the same user id.

Sample input:

```
Nomair Naeem
Ahmad Alrefai
Ten Bradley
Sean Harrap
Marie Harrigan
Yang Tian Zi
```

Sample output:

```
Naeem, Nomair (nnaeem)
Alrefai, Ahmad (aalrefai)
Bradley, Ten (tbradley)
Harrap, Sean (sharrap)
Harrigan, Marie (mharrigan)
Zi, Yang Tian (ytzi)
```

## 4 Programming Question: Big 5

Write the Big 5 for the following classes:

```
class Bintree{
    struct Data{
        // array of integers *on heap*
        int *arr;
        // length of arr
        int datalen;
        // Capacity of arr
        int capacity;

        Data(int cap = 10):arr{(cap == 0) ? nullptr : new int[cap]},
                           datalen{0},
                           capacity{cap}{}

        // Write your big 5 for Data as they appear here.
    };
    Data data;
    Bintree *left;
    Bintree *right;

    Bintree(Bintree *left = nullptr, Bintree *right = nullptr):
        data{},left{left},right{right}{}

    // Write your big 5 for Bintree as they appear here.
};
```

## 5 Conceptual Questions

1. What does the `static` keyword do? List all of uses of `static` keyword you have seen so far (including what you have seen in CS 136).
2. What are the benefits of separate compilation?
3. What is a function's signature?
4. How does `istream` support using the same operator (`>>`) to read in integers and strings?
5. What is the Member Initialization List? Do we have to use it in some cases? List those cases if there are any; if not, explain why.
6. What is an invariant? What is its significance in designing programs?
7. What is the scope resolution operator?

## 6 Optional Conceptual Questions

**Those questions are considered optional since they might be covered in some sections but not others.**

1. What is the modulo bias problem when generating a random integer? What are the solutions C++11 proposed to address it?
2. What are the partial assignment and the mixed assignment problems? What are the solutions to them?
3. Why does C++ standard library wrap all built-in functions and variables into the `std` namespace?
4. Draw a UML for the following description. Identify the classes, potential methods and relationships in between the classes:

The WatCard Office is designing a new university members management system. There are three types of members: Staff, Graduate Students and Undergraduate Students. Each member has a WatIAM ID, a student ID, and a library access code. Staff can be temporary or permanent. Students can be exchange students or regular students, and they can be in any program. Graduate students may have a supervisor (which is a permanent staff). Every member has a digital wallet within the WatCard, which they can add money and spend in supported stores (i.e. only stores can remove money from digital wallets); in addition, undergraduate students can be enrolled in a meal plan, which grants a 30% discount at those stores.