Introduction

General issues

1. Make sure you have the required Java and C programming skills. Discuss it with your tutor if unsure.

2. You are expected to
   - Attend the prac. sessions, and actively participate in the class
   - Keep noise down at all times
   - Follow the announcements and discussions on Canvas

3. Acceptable coding practices in this course include
   - Using sensible and unified naming convention
   - Supporting your code with meaningful descriptions
   - Using modular program structure when appropriate
   - Observing the 80 columns limit
   - No hardcoded or literal value in the source code – use constants instead
   - No submitting uncommented code
   - Handling errors properly and sensibly

4. Have a question? You should be trying to ask it in the following order:
   - Ask during the lecture
   - Ask your tutor during the prac. session
   - Ask it on Canvas → Discussion Forum
   - Attend the consultation times

5. Need to contact someone? Refer to Canvas → Staff Information

Activity: prepare for the course

1. Confirm that your CSIT accounts are functional.

2. Ensure that you are able to
   - log into Moonshot
   - work with the Unix shell
   - access your home directory in the lab
   - access the Blackboard Discussion Forum
   - use the Java API
   - edit, compile and run Java and C Programs on Moonshot

3. You can use your favourite Java IDE to develop your programs, but you will have to demonstrate your program on Moonshot.

Network Programming
Lab 1-3

Aims

- To familiarize students with the environment
- Practice working with character and byte streams in Java

Tasks

Use buffered input and output streams in the following tasks when possible.

1. Using the Java class `InputStream` and `OutputStream`, write a program that takes a sequence of characters from the console, replaces each whitespace character with the underscore '_' character and outputs the changed text to the screen.
2. Write programs that write files and checksums and read them back, as follows.
   a. Write a Java program that takes lines of characters from the console and writes them into a file. The input should stop when the read line contains a single 'x' character only. The program should also calculate the checksum of the whole input, and write it into a different file.
   b. Write another program that reads a character file (e.g. the file generated in the previous part of this question), prints its content to the screen and calculates the checksum of the file. The program should also read the checksum from another file and print it out to the screen (for comparison with the calculated value).

   Hints
   - Look at `CheckedOutputStream, CheckedInputStream`
   - For the checksum variable, use `new Adler32()` or `new CRC32()` in the constructor of the checked streams
   - You need to flush the written output stream before you can access its checksum

Pay attention to the following.

Your program should

- work correctly and handle all exceptions
- terminate gracefully
- use the most appropriate classes for the tasks
- be well commented.