University of Eswatini

Department of Computer Science

CSC411 - Integrative Programming Technologies

MINI PROJECT

Instructions:

This project by a team that consists of two (2) students.

You may use a programming language of your choice.

PART ONE: THE PRODUCER CONSUMER PROBLEM [70 MARKS]

Implement the Producer-Consumer Problem.

The Producer

The producer must produce data that contains student information and store the data in a *ITstudents* class.

The member variables of the *ITstudent* class must be generated using a random generating algorithm. The variables include Student Name, Student ID (8 digits), Programme, a list of Courses and the associated mark for each course.

The producer must wrap the student information into XML format. The names of the XML files shall be captioned with numbers between 1-10. For example, *student1.xml.*

The producer will place the XML file in directory it shares with the buffer. Each time the producer shares information with the buffer, it must insert an integer that corresponds to the file name to the buffer/queue. For example, if the file added is *student1.xml,* 1 must be inserted to the buffer.

The Consumer

The consumer reads the content of the xml file it shares with the buffer.

The consumer must unwrap an XML file and gather the XML file student information into a *ITstudent* class.

The consumer must clear the content of the xml file (or delete the file). Each time the consumer removes/clears an XML file. The consumer must also remove the integer from the buffer.

On the *ITstudent* class, the consumer must calculate the average mark based on the marks allocated to the courses. The consumer must determine whether the student passed or failed. The pass mark is 50%.

The consumer must print on the screen the following: the student’s name, Student ID, Programme, Courses and the associated marks, the average, and the Pass/Fail information.

The Buffer

The buffer is a shared common finite container with a maximum size of 10 elements.

The rules for access to the buffer are as follows:

1. The Producer process should not produce any data when the shared buffer is full.
2. The Consumer process should not consume any data when the shared buffer is empty.
3. The access to the shared buffer should be mutually exclusive i.e., at a time only one process should be able to access the shared buffer and make changes to it.

You may use semaphores to ensure that the above rules are enforced.

Allocation of Marks:

Wrapping and Unwrapping XML data	*[30]*
Implementation of concurrency, synchronization, and other aspects	*[20]*
Run of the producer/consumer problem	*[20]*

PART 2: GITHUB [20 MARKS]

Use GitHub as Git a version control environment to collaborate and share your project with your partner. The source code and other supporting documents, e.g., Readme file, of the project must available be in GitHub.

Each project member must have a GitHub account. A GitHub URL link to your account profile must be submitted. Submit a GitHub URL link to the project.

Allocation of Marks:

Opening the GitHub accounts.	*[5]*
Using the GitHub version control.	*[10]*
Uploading the required files to GitHub.	*[5]*

PART 3: SOCKET PROGRAMMING (EXTRA MARKS) [30 MARKS]

Use socket programming to implement the producer and consumer problem. *[30]*

Submission:

The submission date is **24 July 2023 at 23h00.**

Submit your assignment in a **PDF** file format.

Submit via the Moodle portal.

No deadline extensions will be made.

Only one member may submit the tasks. The names and student numbers the two team members must be written in the submitted file.

Demo Event:

Students will have to demonstrate their tools within a 10–15-minute slots on **Wednesday, 26 July 2023**.