

**ATHLONE INSTITUTE OF TECHNOLOGY**

**SCHOOL OF ENGINEERING**

**SEMESTER 1 EXAMINATIONS 2016**

**December Session**



**BACHELOR OF SCIENCE IN SOFTWARE DESIGN (CLOUD/GAME DEVELOPMENT)**

**YEAR 3**

**OPERATING SYSTEMS & CONCURRENCY**

**External Examiner(s):**

**Mr Jerh O'Connor  
Dr Steven Davy**

**Internal Examiner(s):**

**Dr Sheila Fallon**

**Instructions to candidates:**

Read all questions carefully.

All questions carry equal marks.

Answer **Three** out of **Four** questions.

**Time Allowed: 2 Hrs**

**No. of pages including cover sheet: 3**

Q.1. (a) Define the following terms: Program, Process and Program Counter. (6 marks)

(b) Briefly explain the difference between process scheduling and dispatching. Give two classifications of processes and explain how a Processing Scheduling algorithm should treat the two types of process. (8 marks)

(c) Briefly describe the difference between a process and a thread. Explain **three** reasons why developers use threads and concurrency. (6 marks)

**[20 marks]**

Q.2 (a) Compare and contrast the two ways of creating and starting threads in Java. Give example code to demonstrate your answer. Explain why a programmer may choose one way of creating threads over the other. (7 marks)

(b) *“Even the most modest of systems can benefit from considering **performance** and potential **scalability** in their design”*. With reference to this statement explain the difference between performance and scalability. (6 marks)

(c) State and explain Amdahl's law. Using Amdahl's law, calculate the max speedup with ten processors, for a program with 10% serialization. Would it be efficient to use 100 processors for this program? Explain your answer. (7 marks)

**[20 marks]**

Q.3 (a) Briefly explain the terms Critical Section and Mutual Exclusion. The monitor is a key mechanism in Java for ensuring mutual exclusion. Explain how it works. (5 marks)

(b) Explain the Producer Consumer design pattern. Describe **four** benefits of this design pattern (5 marks)

(c) Using short sections of java code illustrate  
(i) how a producer thread creates items and inserts them into an instance of `java.util.concurrent.BlockingQueue`.  
(ii) how a consumer thread removes items from the `BlockingQueue`  
(iii) a `main()` method that creates and starts a producer thread **and** a consumer thread. (10 marks)

**[20 marks]**

- Q.4. (a) Explain how the `java.util.concurrent` package separates task submission and task execution. In particular what are the relevances of the interfaces `Executor` and `ExecutorService`?  
(6 marks)
- (b) Explain how “Callable”s and “Future”s are used to execute tasks when we want future access to the results of the task.  
(7 marks)
- (c) Software Transactional Memory (STM) is an alternative to lock-based synchronization. Briefly explain the STM approach to concurrency including the benefits and overheads of STM.  
(7 marks)
- [20 marks]**