



UNIVERSITY OF LAGOS  
DEPARTMENT OF COMPUTER SCIENCES  
B.Sc. (Hons) DEGREE EXAMINATIONS DECEMBER 2022  
FIRST SEMESTER 2020/2021 SESSION  
CSC302: CONCURRENT PROGRAMMING

**Instruction:** Answer question one (1) and two (2) and any other two (2) questions **Time Allowed: 2 hours**

- 1 Search Engine Results Pages (SERP) are the pages displayed by search engines in response to a query by a user. Each result displayed normally includes a title, a link that points to the actual page on the Web, and a short description showing where the keywords have matched content within the page for organic results. The SERP are ranked based on relevance for organic results. Considering the semantic search of users' query gives more accurate SERP. More importantly, summarizing the relevant content of SERP for users instead of the displayed titles and links will be more useful to users.

Show your design as well as the appropriate program snippets that show your multithreaded program solutions for returning summarized relevant content of the SERP.

- 2 a Discuss three (3) thread-safety issues in concurrent programs that developers should be aware of.  
b What is the difference when lock is applied to an instance method and when lock is applied to a static method?  
c Use the program listing below to answer the following:  
i Identify the shared resource in the code listing  
ii What is the function of the method *newCachedThreadPool()* in the listing and how different is the method from *newFixedThreadPool()*?  
iii Do you think the code will yield desirable output? Justify your response.  
iv Use Java lock interface to resolve the possible conflict in the code listing.  
v Use *synchronized* keyword to resolve the possible conflict in the code listing.  
vi Use semaphore to control the number of threads that access the shared resource.

```
import java.util.concurrent.*;

public class Exercise1 {
    private static Account acc = new Account();
    public static void main(String[] args) {
        ExecutorService executor = Executors.newCachedThreadPool();
        for (int i = 0; i < 100; i++) { executor.execute(new AddMoney()); }
        executor.shutdown();
        while (!executor.isTerminated()) {
        } System.out.println("What is balance ? " + acc.getBalance());
    }
    private static class AddMoney implements Runnable {
        public void run() {
            acc.deposit(1);
        }
    }
    private static class Account {
        private int balance = 0;
        public int getBalance() { return balance; }
        public void deposit(int amount) {
            int newBalance = balance + amount;
            try {
                Thread.sleep(10);
            }
            catch (InterruptedException ex) { }
            balance = newBalance;
        }
    }
}
```

- 3
  - a Explain how deadlock, livelock and starvation can occur with respect to the "dining philosophers" problem.
  - b By using example, differentiate between simple lock-ordering deadlock and dynamic lock-ordering deadlock. Write code snippet that resolves each of these deadlocks.
- 4
  - a Define the term synchronizer.
  - b Compare and contrast the following synchronizers: latches, barrier and semaphore
  - c Give two applications of semaphore.
  - d
    - i. What is a synchronized collection?
    - ii. Is ArrayList synchronized? Justify your answer.
- 5
  - a Designing a thread-safe class is about managing concurrent access to the same shared mutable state variable in the class. Explain each of the underlined term.
  - b List and explain four (4) ways of fixing a broken multithreaded program.
  - c Study the following codes carefully. State whether each section of the codes is thread safe or not. Justify your answers and *if need be*, make it thread safe.

```
public class Task1{
    private int v;
    public int getNext() {
        return v++;
    }
}
```

```
public class Task2 implements Servlet{
    private long count = 0;
    public long getCount() {return count;}
    public void service( ServletRequest req, ServletResponse resp) {
        BigInteger i = extractFromRequest(req);
        BigInteger[ ] factors = factor(i);
        ++count;
        encodeIntoResponse(resp, factors); }}
}
```

```
public class Task3 implements Servlet{
    public void service( ServletRequest req, ServletResponse resp) {
        BigInteger i = extractFromRequest(req);
        BigInteger[ ] factors = factor(i);
        encodeIntoResponse(resp, factors); }}
}
```

- 6
  - a Consumer/producer example is used to illustrate communication among threads. In Java, Conditions are used to facilitate this communication. State how condition is created and how its methods can be used for threads communication in relation to the consumer/producer scenario.
  - b Use the fork/join framework to describe parallel implementation of a merge sort algorithm