ii. Linked –list(using linked-list)
iii. Indirect allocation (indexing)

5. Implementation of contiguous allocation techniques:
   i. Worst-Fit
   ii. Best- Fit
   iii. First- Fit
6. Calculation of external and internal fragmentation
   i. Free space list of blocks from system
   ii. List process file from the system
7. Implementation of compaction for the continually changing memory layout and calculate total movement of data
8. Implementation of resource allocation graph RAG)
9. Implementation of Banker‟s algorithm
10. Conversion of resource allocation graph (RAG) to wait for graph (WFG) for each type of method used for storing graph.
11. Implement the solution for Bounded Buffer (producer-consumer)problem using inter process communication techniques-Semaphores
12. Implement the solutions for Readers-Writers problem using inter process communication technique - Semaphore

BCS452- Object Oriented Programming with Java

List of Experiments (Indicative & not limited to)

1. Use Java compiler and eclipse platform to write and execute java program.
2. Creating simple java programs using command line arguments
3. Understand OOP concepts and basics of Java programming.
4. Create Java programs using inheritance and polymorphism.
5. Implement error-handling techniques using exception handling and multithreading.
6. Create java program with the use of java packages.
7. Construct java program using Java I/O package.
8. Create industry oriented application using Spring Framework.
9. Test RESTful web services using Spring Boot.
10. Test Frontend web application with Spring Boot

BCS453- Cyber Security Workshop

List of Experiments (Indicative & not limited to)

Module 1: Packet Analysis using Wire shark

1. Basic Packet Inspection: Capture network traffic using Wire shark and analyze basic protocols like HTTP, DNS, and SMTP to understand how data is transmitted and received.

2. Detecting Suspicious Activity: Analyze network traffic to identify suspicious patterns, such as repeated connection attempts or unusual communication between hosts.
3. Malware Traffic Analysis: Analyze captured traffic to identify signs of malware communication, such as command-and-control traffic or data infiltration.
4. Password Sniffing: Simulate a scenario where a password is transmitted in plaintext. Use Wireshark to capture and analyze the packets to demonstrate the vulnerability and the importance of encryption.
5. ARP Poisoning Attack: Set up an ARP poisoning attack using tools like Ettercap. Analyze the captured packets to understand how the attack can lead to a Man-in-the-Middle scenario.

Module 2: Web Application Security using DVWA

1. SQL Injection: Use DVWA to practice SQL injection attacks. Demonstrate how an attacker can manipulate input fields to extract, modify, or delete database information.
2. Cross-Site Scripting (XSS): Exploit XSS vulnerabilities in DVWA to inject malicious scripts into web pages. Show the potential impact of XSS attacks, such as stealing cookies or defacing websites.
3. Cross-Site Request Forgery (CSRF): Set up a CSRF attack in DVWA to demonstrate how attackers can manipulate authenticated users into performing unintended actions.
4. File Inclusion Vulnerabilities: Explore remote and local file inclusion vulnerabilities in DVWA. Show how attackers can include malicious files on a server and execute arbitrary code.
5. Brute-Force and Dictionary Attacks: Use DVWA to simulate login pages and demonstrate brute-force and dictionary attacks against weak passwords. Emphasize the importance of strong password policies.