

**FINAL EXAMINATION**

**BACHELOR OF INFORMATION TECHNOLOGY (HONOURS) IN COMPUTER
APPLICATION DEVELOPMENT**

**BACHELOR OF INFORMATION TECHNOLOGY (HONOURS) IN CYBER
SECURITY**

**BACHELOR OF INFORMATION TECHNOLOGY (HONOURS) IN (DATA
ANALYTICS)**

**COURSE : ALGORITHMIC DATA STRUCTURES/
DATA STRUCTURES & ALGORITHMS**

COURSE CODE : SWC3524

DURATION : 3 HOURS

INSTRUCTIONS TO CANDIDATES :

1. This question paper consists of **TWO (2)** parts. : PART A (30 questions)
: PART B (4 questions)
2. Answer ALL questions in the Answer Booklet provided.
3. Please check to make sure that this examination pack consists of :
 - i. The Question Paper
 - ii. An Answer Booklet
4. Do not bring any material into the examination-hall. The use of calculator is allowed.
5. Please write your answer using permanent ink.

MYKAD/ PASSPORT NO. : _____
ID. NO. : _____
LECTURER : _____
SECTION : _____

DO NOT OPEN THIS QUESTION PAPER UNTIL YOU ARE TOLD TO DO SO

This question paper consists of 16 printed pages including the front page

CONFIDENTIAL

PART A : MULTIPLE CHOICE

Instruction: Choose the **CORRECT** answer.

1. What is the main purpose of writing pseudocode before developing a program?
 - A. To write complete Java syntax
 - B. To record all programming errors
 - C. To compile the program before execution
 - D. To describe algorithm steps using simple and readable language

2. In Object-Oriented Programming, **encapsulation** refers to?
 - A. Creating objects with the same attributes
 - B. Declaring several classes with the same name
 - C. Redefining inherited methods from a superclass
 - D. Protecting internal data and allowing access through selected methods

3. What is the primary purpose of a flowchart in programming?
 - A. Storing data in a structured format
 - B. Determining the runtime of algorithms
 - C. Writing the source code for Java programs
 - D. Representing the execution flow of an algorithm graphically

4. Which Java keyword indicates that a method does not return any value?
 - A. `int`
 - B. `void`
 - C. `public`
 - D. `static`

5. Consider the following recursive implementation of the factorial function:

```
public class Factorial {  
    public static int calculateFactorial(int number) {  
        if (number == 0) {  
            return 1;  
        } else {  
            return number * calculateFactorial(number - 1);  
        }  
    }  
}
```

What value will be returned when `calculateFactorial(3)` is executed?

- A. 1
 - B. 3
 - C. 5
 - D. 6
6. What is the purpose of defining a class in Java?
- A. To run statements that are not related to each other
 - B. To open and execute external programs automatically
 - C. To store only temporary values during program execution
 - D. To combine related attributes and methods into one blueprint
7. Which statement **BEST** explains the function of a selection structure in programming?
- A. It runs a block of code repeatedly without checking any condition
 - B. It executes instructions in a fixed order only
 - C. It allows the program to choose an action based on whether a condition is true or false
 - D. It stores several values in one variable

8. Consider the following classes in Java:

```
class Animal {
    public void makeSound() {
        System.out.println("Animal makes a sound");
    }
}
class Cat extends Animal {
    @Override
    public void makeSound() {
        System.out.println("Cat meows");
    }
}
class Dog extends Animal {
    @Override
    public void makeSound() {
        System.out.println("Dog barks");
    }
}
public class TestAnimal {
    public static void main(String[] args) {
        Animal a1 = new Cat();
        Animal a2 = new Dog();

        a1.makeSound();
        a2.makeSound();
    }
}
```

What will be the output when the `main()` method is executed?

- A. Cat meows
Dog barks
 - B. Dog barks
Cat meows
 - C. Cat meows
Animal makes a sound
 - D. Animal makes a sound
Animal makes a sound
9. Which statement **BEST** explains the connection between a class and an object in Java?
- A. An object stores inheritance rules for a class
 - B. An object is used as a blueprint to create several classes
 - C. A class only exists after an object is created during program execution
 - D. A class defines the structure, while an object is a real instance created from it

10. Which of the following statement **BEST** describes an array in Java?
- A. Arrays are dynamic in size by default
 - B. Arrays automatically sort their elements
 - C. Arrays store elements of the same data type
 - D. Arrays can store multiple values of the same data type using one variable name
11. Which of the following classes in Java is **synchronized** by default?
- A. Vector
 - B. LinkedList
 - C. ArrayList
 - D. HashMap
12. Which feature of the Java Collections Framework helps improve code reusability?
- A. It only supports a small number of collection classes
 - B. It removes the use of interfaces in Java programming
 - C. It requires programmers to create all data structures from scratch
 - D. It provides common interfaces and ready-to-use classes for managing collections
13. Why it is important to select the correct data structure when developing a program?
- A. It allows the program to ignore security controls
 - B. It reduces the time needed to compile the source code
 - C. It increases the amount of data automatically without limits
 - D. It helps the program run more efficiently based on the task requirements
14. A program needs a collection that allows **duplicate values, permits multiple null values**, and **supports fast access by index**. Which collection would be the most suitable?
- A. HashSet
 - B. ArrayList
 - C. TreeMap
 - D. LinkedList

15. A courier service needs to process delivery requests based on the order in which they are received. Which data structure is most suitable for this situation?

- A. Stack
- B. Queue
- C. ArrayList
- D. PriorityQueue

16. What happens when `remove()` method is used to dequeue an element from an empty Queue implemented using `LinkedList` in Java?

- A. It returns -1
- B. It returns null
- C. It throws a `NullPointerException`
- D. It throws a `NoSuchElementException`

17. Given the following operations on a stack:

```
Stack<Integer> stack = new Stack<>();  
stack.push(8);  
stack.push(12);  
stack.push(18);  
stack.pop();  
stack.push(25);  
System.out.println(stack.peek());
```

What will the `peek()` method return?

- A. 5
- B. 12
- C. 19
- D. 25

18. A circular queue is implemented using an array with a size of 6. The front pointer is currently at index 2, and the rear pointer is at index 1. Which operation will cause an overflow condition?

- A. Checking the element at the front
- B. Displaying all elements in the queue
- C. Removing an element from the queue
- D. Inserting a new element into the queue

19. What happens when an `ArrayList` becomes full and a new element is added?

- A. It converts itself into a fixed-size array.
- B. It stops the program and displays an error.
- C. It deletes the first element to create extra space.
- D. It expands its internal storage automatically to fit more elements.

20. Which set of operations is commonly used in a `Stack` data structure?

- i. `push(E element)`
- ii. `pop()`
- iii. `peek()`
- iv. `isEmpty()`

- A. i and ii only
- B. ii and iii only
- C. All of the above
- D. i, ii, and iv only

21. What will be the **OUTPUT** of the following code?

```
ArrayList<Integer> values = new ArrayList<>();  
values.add(3);  
values.add(6);  
values.add(9);  
values.add(1, 12);  
values.set(2, 15);  
System.out.println(values);
```

- A. `[3, 6, 15, 9]`
- B. `[3, 6, 9, 15]`
- C. `[3, 15, 6, 9]`
- D. `[3, 12, 15, 9]`

22. What happens if the `pop()` method is executed on an empty `Stack` in Java?

- A. It returns null
- B. It returns -1
- C. It throws a `NullPointerException`
- D. It throws an `EmptyStackException`

23. What will be the state of the stack after the following code is executed?

```
Stack<Integer> stack = new Stack<>();  
  
for (int i = 2; i <= 10; i += 2) {  
    stack.push(i);  
}  
  
while (stack.size() > 3) {  
    stack.pop();  
}  
  
System.out.println(stack);
```

- A. [2, 4]
B. [8, 10]
C. [2, 4, 6]
D. [2, 4, 6, 8, 10]
24. A printer receives several print jobs from users. The first document sent to the printer will be printed first. Which principle **BEST** describes this data structure?
- A. Doubly Ended
B. Random Access
C. Last In First Out (LIFO)
D. First In First Out (FIFO)
25. Which of the following method stores collided keys in a Linked List at the same hash table index?
- A. Merge Sort
B. Binary Search
C. Separate Chaining
D. Depth-First Search
26. Consider a hash table of size 13 and a hash function defined as $h(k) = k \bmod 11$. Given a set of keys {22, 15, 33, 28, 44}, how many collisions occur when these keys are inserted using separate chaining?
- A. 0
B. 1
C. 2
D. 3

27. A compiler uses a hash table to store information about symbols in a program. Given that the number of different possible variable names is vastly larger than the expected number of names used in a real-world program, what is the **BEST** strategy for choosing the size of the hash table?
- A. Choose a random size to avoid predictable patterns
 - B. Choose a small table size since collisions will be rare
 - C. Use a table size equal to the maximum possible number of symbols
 - D. Use a prime number that is roughly twice the number of expected symbols
28. Consider the following code snippet. What will be the Output when the code is executed?

```
import java.util.ArrayList;
import java.util.Stack;

public class Main {
    public static void main(String[] args) {
        ArrayList<String> items = new ArrayList<>();
        Stack<String> stack = new Stack<>();

        items.add("P");
        items.add("Q");
        items.add("R");

        stack.push("10");
        stack.push("20");
        stack.push("30");
        items.add(2, "S");
        stack.pop();
        items.remove("Q");

        String topStack = stack.peek();

        System.out.println(items);
        System.out.println(topStack);
    }
}
```

- A. [P, S, R] and 30
- B. [P, S, R] and 20
- C. [P, R, S] and 10
- D. [P, Q, S, R] and 20

29. Which probing method adds checks the next slot by adding a fixed step value after a collision?
- A. Linear Probing
 - B. Double Hashing
 - C. Quadratic Probing
 - D. Separate Chaining
30. Given a hash table of size 12 and a hash function $h(k) = k \% 12$, what is the hash index for key 38?
- A. 1
 - B. 2
 - C. 4
 - D. 6

(TOTAL: 30 MARKS)

PART B : STRUCTURED QUESTIONS

Instruction: Answer ALL questions.

QUESTION 1

- a. List **FOUR (4)** main principles of Object-Oriented Programming (OOP). (4 marks)
- b. Given the following recursive function and Java program to calculate b^e , where b is the base and e is the exponent:

$$\text{power}(b, e) = \begin{cases} 1, & \text{if } e = 0 \\ b \times \text{power}(b, e - 1), & \text{if } e > 0 \end{cases}$$

```
public class PowerCalculator {  
    // Recursive method to calculate b^e  
    public static int power(int b, int e) {  
        if (e == 0) // Base case  
            return 1;  
        else  
            return b * power(b, e - 1); // Recursive case  
    } // end of recursive method  
  
    public static void main(String[] args) {  
        int base = 3, exponent = 3;  
        System.out.println(power(base, exponent));  
    } // end of main method  
} // end of driver class
```

If you are calculating 3^3 using the above recursive method, demonstrate how each recursive call works by tracing the function.

(8 marks)

c. Consider the following Java classes:

```
class Vehicle {
    void move() {
        System.out.println("Vehicle is moving");
    }
}

class Car extends Vehicle {
    void move() {
        System.out.println("Car is driving");
    }
}

class Bicycle extends Vehicle {
    void move() {
        System.out.println("Bicycle is cycling");
    }
}

class Boat extends Vehicle {
    void move() {
        System.out.println("Boat is sailing");
    }
}

public class TestVehicle {
    public static void main(String[] args) {
        Vehicle v;
        v = new Car();
        v.move();

        v = new Bicycle();
        v.move();

        v = new Boat();
        v.move();
    }
}
```

Give the **OUTPUT** when the `main()` method is executed?

(3 marks)

(Total: 15 marks)

QUESTION 2

- a. Explain what is the meaning of meant Java Collections Framework (JCF) and describe how it helps programmers manage data in Java programs.
(4 marks)
- b. Draw a simple hierarchy diagram to show how the `List` interface is implemented by its common Java classes. Your diagram should clearly show the relationship between `List`, `ArrayList`, `LinkedList`, and `Vector`.
(3 marks)
- c. The `LinkedList` Abstract Data Type (ADT) provides a collection of methods methods that help programmers manage data efficiently, especially when inserting, removing, or searching for elements in a `list`. Describe the functionality of the following methods and provide a clear Java code example to show how each method works:
- i. `addFirst(E element)`
(4 marks)
- ii. `indexOf(object o)`
(4 marks)
- (Total: 15 marks)

QUESTION 3

- a. Explain how insertion operations are performed in a Stack and a Queue. Support your explanation with appropriate examples and simple diagrams. (6 marks)
- b. Illustrate the complete execution of the CourseRegistrationManager program provided below.

Based on the given Java code, draw sequential diagrams to show the exact node structures and states of the LinkedList as it undergoes the insertion, update, and deletion operations. You may represent the nodes as standard Singly LinkedList nodes.

```
import java.util.LinkedList;

class Course {
    String courseCode;
    String courseName;

    public Course(String courseCode, String courseName) {
        this.courseCode = courseCode;
        this.courseName = courseName;
    } // end of constructor

    @Override
    public String toString() {
        return "Course Code: " + courseCode + ", Course Name: "
+ courseName;
    } // end of method
} // end of class

public class CourseRegistrationManager {
    public static void main(String[] args) {
        LinkedList<Course> courses = new LinkedList<>();

        // Adding four courses to the LinkedList
        courses.add(new Course("AIC2013", "Applied Artificial
Intelligence"));
        courses.add(0, new Course("SWC3524", "Data
Structures"));
        courses.add(new Course("DBS3013", "Database Systems"));
        courses.addFirst(new Course("WEB2023", "Web
Programming"));
        // Updating the name of the 3rd course
        courses.get(2).courseName = "Applied Artificial
Intelligence (Updated)";
    }
}
```

(10 marks)

```
// Removing a course from the LinkedList using for-loop
String courseToRemove = "SWC3524";

    for (int i = 0; i < courses.size(); i++) {

        if(courses.get(i).courseCode.equals(courseToRemove)) {
            courses.remove(i);
            break;
        } // end of if
    } // end of for
} // end of main method
} // end of class
```

- c. Given the following Java program that implements a `LinkedList` to manage a list of appointments and a `Queue` for customer service tickets, trace the execution of the program. Based on the given Java code below, display the Output after executing the program.

```
import java.util.LinkedList;
import java.util.Queue;

public class DataStructureTrace {
    public static void main(String[] args) {
        // Initialize LinkedList for appointments and Queue for
        // service tickets
        LinkedList<String> appointments = new LinkedList<>();
        Queue<String> serviceTickets = new LinkedList<>();

        appointments.add("Medical Checkup");
        appointments.add("Dental Review");
        appointments.addFirst("Registration");

        serviceTickets.offer("Ticket101");
        serviceTickets.offer("Ticket102");
        // Display selected elements
        System.out.println("First appointment: " +
            appointments.getFirst());
        System.out.println("First service ticket: " +
            serviceTickets.peek());
        System.out.println("Second appointment: " +
            appointments.get(1));
        System.out.println("Processed service ticket: " +
            serviceTickets.poll());

    } // end of main method
} // end of class
```

(4 marks)

(Total: 20 marks)

QUESTION 4

- a. Describe how a hash function is used to store and retrieve data in a hash table. Explain why hashing is useful for improving data access performance. (4 marks)
- b. Given a hash table of size 7 with the hash function defined as $\text{hash}(\text{key}) = \text{key} \% 7$. You are required to insert the following keys into the hash table: 10, 21, 14, 17, 24.
- i. Illustrate how to resolve any collisions that occur during the insertion process using Open Addressing (Linear Probing). (5 marks)
- ii. Provide a final Output of the hash table after all keys have been inserted. (5 marks)
- c. Explain **TWO (2)** advantages of using a hash table for storing and retrieving data compared to arrays or linked lists. Support your answer with suitable examples. (6 marks)

(Total : 20 marks)

(TOTAL: 70 MARKS)

(TOTAL :100 MARKS)

END OF QUESTION PAPER