NOIDA INSTITUE OF ENGINEERING & TECHNOLOGY GREATER NOIDA

2st ASSESSMENT

OBJECT ORIENTED TECHNIQUES
USING JAVA[**BCSE0352**]

Total Marks:30
Section A Quize:10X1=10
Section B Marks :4X5=20
Date:25/09/2024

Notes: Attempt Any four questions, all carries equal marks

* After completing all questions in Section B,
Open your MS Office and write your:

a) Name
b) Roll No
c) Branch
d) Section
e) Workshop Lab Number

Then, put all four program source codes along with their output screenshots and convert it into a PDF file named as NAME_BRANCH_SECTION.

*Upload this PDF file in MS Forms

Section B

QUESTION 1: Write a Java program to demonstrate Method Overriding and Exception Handling using the following rules:

Instructions:

- The **superclass method** throws a **checked exception**, and the **subclass method** overrides it, either throwing the same exception or no exception.
- The **superclass method** does **not throw any exception**, and the **subclass method** overrides it without throwing a checked exception but may throw an unchecked exception.

Your program should include:

- A superclass **SuperClass** with:
  a. A method **showMessage()** that throws a checked exception (Exception).
  b. A method **display()** that does not throw any exception.
- A subclass **SubClass** that:
  a. Overrides **showMessage()** to throw the same checked exception.
  b. Overrides **display ()** to throw an unchecked exception (RuntimeException).
- A main () method that creates objects of the superclass and subclass, demonstrating exception handling using try-catch blocks for both checked and unchecked exceptions.

OR

QUESTION 1: Write a Java program to demonstrate Polymorphism and Exception Handling using the following guidelines:

Instructions:

- The superclass method throws no exception, and the subclass method overrides it without throwing an exception.
- Another subclass method throws an unchecked exception and is handled using try-catch in the main method.

Your program should include:

- A superclass Animal with:
  1. A method sound() that prints a generic message and throws no exception.
- A subclass Dog that:
  1. Overrides sound() to print a specific message like "Dog barks" without throwing any exception.
  2. Adds a new method fetch() that throws an unchecked exception (e.g., IllegalStateException).
- A main() method that:
  1. Demonstrates polymorphism by calling sound() using both superclass and subclass objects.

2. Uses try-catch to handle the unchecked exception thrown by fetch().

QUESTION 2: Write a Java program to perform matrix addition on two two-dimensional arrays. The program should follow these steps:

Instructions:

a) Input:

- Use the Scanner class to allow the user to input the dimensions (rows and columns) and elements of two matrices.
- Ensure that both matrices have the same dimensions (same number of rows and columns). If the dimensions don't match, display an error message and terminate the program.

b) Matrix Addition:

- Add the two matrices element-wise. The resulting matrix's dimensions should be the same as the input matrices.
- Write a method to perform matrix addition and return the resulting matrix.

c) Display:

- Display the two input matrices in matrix form.

Enter the number of rows and columns for both matrices: Rows: 2 Columns: 3 Enter elements for Matrix 1: 1 2 3 4 5 6 Enter elements for Matrix 2: 7 8 9 1 2 3	Output Matrix 1: 1 2 3 4 5 6 Matrix 2: 7 8 9 1 2 3 Resulting Matrix after Addition: 8 10 12 5 7 9

OR

QUESTION 2: Write a Java program to perform matrix multiplication on two two-dimensional arrays. The program should implement the following steps:

Instructions:

a) **Input**:

- Use the Scanner class to allow the user to input the dimensions (rows and columns) and elements of two matrices.
- Ensure the matrices can be multiplied, i.e., the number of columns in the first matrix must be equal to the number of rows in the second matrix. If the matrices cannot be multiplied, display an error message and terminate the program.

b) **Matrix Multiplication**:

- Multiply the two matrices according to matrix multiplication rules. The resulting matrix's dimensions should be (rows of the first matrix x columns of the second matrix).
- Write a method to perform matrix multiplication and return the resulting matrix.

c) **Display**:

- Display the two input matrices in matrix form.
- Display the resulting matrix after multiplication.

Example: For matrices A and B: Matrix A (2x3): 1 2 3 4 5 6 Matrix B (3x2): 7 8 9 10 11 12	The result of multiplying A and B should be: Resulting Matrix (2x2): 58 64 139 154

QUESTION 3: Write a Java program that demonstrates the use of lambda expressions and functional interfaces for string operations. Your task is to:

Instructions:

- Define a functional interface that has a single abstract method to perform an operation on two strings.

- Implement various operations (like concatenation, comparison, and finding the longer string) using lambda expressions to provide the implementation of the functional interface.
- Write a method that takes two strings and a lambda expression to perform the corresponding operation.
- Use exception handling to handle any potential null pointer exceptions.
- Display the result of each operation.

OR

Question 3: Write a Java program that demonstrates the use of the StringBuffer class, highlighting its mutability and thread safety features. Your program should include examples of commonly used StringBuffer methods and explain how StringBuffer differs from String in terms of mutability.

Instructions:

a) Create a StringBuffer object:
- Initialize it with a string value.

b) Demonstrate mutability:
- Use the append(), insert(), delete(), and replace() methods to modify the contents of the StringBuffer object, showing that it changes without creating a new object (in contrast to String's immutability).

c) Reverse the content:
- Reverse the content of the StringBuffer using the reverse() method and print the result.

d) Capacity management:
- Display the initial capacity of the StringBuffer and demonstrate how the capacity grows dynamically when new characters are appended.

e) Thread safety:
- Provide a brief explanation in the comments about the thread-safe nature of StringBuffer and how it differs from StringBuilder.

f) Comparison with String:
- Compare String and StringBuffer in terms of mutability by showing how modifying a String creates a new object, while StringBuffer modifies the same object in place.

QUESTION 4: Write a Java program that demonstrates the use of exception handling for multiple exceptions. Your task is to:

Instructions:

a) Define a method that performs division of two integers.
b) Define a method that accesses an element in an array.
c) Define a method that reads a file.
d) Use try-catch blocks to handle ArithmeticException, ArrayIndexOutOfBoundsException, and FileNotFoundException.

e) Use a finally block to display a message indicating that the operation has been attempted.
f) Write a method that calls these methods and handles any exceptions that are thrown.
g) Display the result of each operation or an appropriate error message if an exception occurs.