C-DAC Mumbai

PG-DAC Aug25 Batch (Kharghar)

Re-Lab Exam

Module 2: Object-Oriented Programming with Java

Time: 1.5 hrs **Marks: 40**

Question 1 **[10 Marks]**

Write a Java program with a base class Person and a derived class Student. The Person class should have two data members: name (String) and age (int).

- **Include**:
  - A parameterised constructor to initialise these data members.
  - A displayDetails() method to display the details of the person.

The Student class should inherit from Person and include additional data members:

- rollNo (int)
- marks (double).
- **The Student class should**:
  - Include a parameterised constructor to initialise all data members, including those of the base class.
  - Override the displayDetails() method to display the details of the student, including rollNo and marks.
  - Include a validateMarks() method to ensure the marks are between 0 and 100.
    - If marks are not within this range, print an error message: "Invalid Marks" and reset the marks to 0.

Tasks:

1. Create 3 objects of both the Person and Student classes.
2. Use the validateMarks() method for Student objects.
3. Display the details of all objects using the displayDetails() method.

Requirements:

- Proper class and method structure.
- Use of constructors for initialisation.
- Input validation for marks in the Student class.
- Demonstration of inheritance and polymorphism.
- Clear output displaying the details of Person and Student.

Question 2 **[15 Marks]**

Library System with Custom Exception

Part A: Design a class named Library with the following specifications:

- **Data Members**:
    1. bookList (ArrayList of Book objects).
- **Methods**:
    1. addBook(Book book) to add a book to the list.
    2. removeBook(String title) to remove a book by title. If the book does not exist, throw a custom exception BookNotFoundException.
    3. displayBooks() to display all books in the library.

Part B: Implement the Library class with a Book class having the following data members:

- title (String)
- author (String)
- isbn (String).

Include appropriate constructors, getters, and setters in the Book class.

Tasks:

1. Add 3 books to the library.
2. Display all books using the displayBooks() method.
3. Try to remove a book by title.
4. Handle the BookNotFoundException if the book does not exist and print an appropriate message (e.g., "Book titled [title] not found.").
5. Display all books again after attempting to remove the book.

Requirements:

- Proper class and method structure for Library and Book.
- Use of ArrayList to store books.
- Custom exception handling for non-existent books.
- Display book details in the format:
    - Title: [title], Author: [author], ISBN: [isbn].

Question 3 **[15 Marks]**

File Handling and Word Frequency

Part A: Write a Java program to read a file named input.txt and store each line in an ArrayList. Then, count the frequency of each word in the file using a HashMap and display the results.

- Handle potential exceptions during file operations.

Part B: Modify the program to handle a custom exception named FileEmptyException. This exception should be thrown if the file is empty.

Tasks:

1. Use BufferedReader to read the file and store each line in an ArrayList.
2. Use a HashMap to count the frequency of each word (case-insensitive).
3. If the file is empty, throw a FileEmptyException and display an appropriate message: "The file is empty. No data to process.".
4. Display word frequencies in the format:

    - [word]: [frequency].

Content for input.txt :

Java is a programming language.

Java is widely used.

Programming with Java is fun and productive.

Fun with Java.

Requirements:

- Use of BufferedReader for file reading.
- Store each line in an ArrayList.
- Use HashMap for word frequency counts (case-insensitive).
- Custom exception handling for empty files.