**Java Revision Questions and Projects**

**Easy Questions**

**Strings:**

1. Write a Java program to concatenate two strings.
2. Write a Java program to find the length of a string.
3. Write a Java program to convert a string to uppercase and lowercase.

**Loops:**

1. Write a Java program to print numbers from 1 to 10 using a for loop.
2. Write a Java program to calculate the sum of first 10 natural numbers using a while loop.
3. Write a Java program to reverse a number using a while loop.

**Arrays:**

1. Write a Java program to find the largest and smallest elements in an array.
2. Write a Java program to calculate the sum of all elements in an array.
3. Write a Java program to sort an array in ascending order.

**Functions:**

1. Write a Java program with a function to find the maximum of two numbers.
2. Write a Java program with a function to check if a number is even or odd.
3. Write a Java program with a function that takes a string and returns the reversed string.

**OOP (Access Modifiers and Constructors):**

1. Write a Java program to demonstrate the use of private, protected, and public access modifiers.
2. Write a Java program with a class that includes a constructor to initialize an object.
3. Write a Java program to create multiple constructors in a class.

**Exception Handling:**

1. Write a Java program to handle ArrayIndexOutOfBoundsException.
2. Write a Java program to handle NumberFormatException.
3. Write a Java program with a try-catch block to handle exceptions.

**Medium Questions**

**Strings:**

1. Write a Java program to compare two strings for equality.
2. Write a Java program to find the first occurrence of a character in a string.

**Loops:**

1. Write a Java program to print the multiplication table of a given number using a do-while loop.
2. Write a Java program to check if a number is a prime number using a for loop.

**Arrays:**

1. Write a Java program to find the duplicate elements in an array.
2. Write a Java program to merge two arrays into a third array.

**Functions:**

1. Write a Java program with a function to calculate the factorial of a number.
2. Write a Java program with a function to find the GCD of two numbers.

**OOP (Inheritance):**

1. Write a Java program to demonstrate single inheritance.
2. Write a Java program to override a method in a subclass.
3. Write a Java program to use the super keyword to call a superclass constructor.

**OOP (Polymorphism):**

1. Write a Java program to demonstrate method overloading.
2. Write a Java program to use dynamic method dispatch (runtime polymorphism).
3. Write a Java program to create a superclass reference to a subclass object.

**Exception Handling:**

1. Write a Java program to demonstrate the use of finally block.
2. Write a Java program to create and throw a custom exception.

**Hard Questions**

**Strings:**

1. Write a Java program to implement your own version of the **substring** method.

**Loops:**

1. Write a Java program to print all prime numbers up to a given number using nested loops.

**Arrays:**

1. Write a Java program to implement the quicksort algorithm.

**Functions:**

1. Write a Java program with a recursive function to solve the Tower of Hanoi problem.

**OOP (Abstract Classes and Interfaces):**

1. Write a Java program to create an abstract class with an abstract method and a concrete method.
2. Write a Java program to implement an interface.
3. Write a Java program to create a class that implements multiple interfaces.
4. Write a Java program to use an abstract class and an interface together.

**OOP (Polymorphism):**

1. Write a Java program to demonstrate the use of the **instanceof** operator.

**Exception Handling:**

1. Write a Java program to handle multiple exceptions in a single try-catch block.

**Beginner Level Projects:**

**1. Calculator**

**Description:** Create a basic calculator that can perform simple arithmetic operations like addition, subtraction, multiplication, and division.

**Features:**

* Command-line interface to take input.
* Functions for each arithmetic operation.
* Error handling for division by zero and invalid input.

**2. To-Do List**

**Description:** Develop a console-based to-do list application where users can add, remove, and view tasks.

**Features:**

* Add new tasks.
* Mark tasks as complete.
* Delete tasks.
* Display all tasks with their completion status.

**3. Library Management System**

**Description:** Implement a simple library management system to keep track of books and users.

**Features:**

* Add new books.
* Issue books to users.
* Return books.
* View list of available books.

**4. Simple Banking System**

**Description:** Create a basic banking application that allows users to create accounts, deposit and withdraw money, and check balances.

**Features:**

* Create new accounts.
* Deposit and withdraw funds.
* Check account balance.
* Transfer money between accounts.

**5. Student Grade Management System**

**Description:** Develop an application to manage student records, including their grades.

**Features:**

* Add new student records.
* Update student grades.
* Display student information.
* Calculate the average grade for each student.

**6. Contact Management System**

**Description:** Build a program to manage contacts with names, phone numbers, and email addresses.

**Features:**

* Add new contacts.
* Search contacts by name.
* Update contact information.
* Delete contacts.
* Display all contacts.

**7. Simple Chat Application**

**Description:** Create a simple console-based chat application for two users.

**Features:**

* Users can send and receive messages.
* Implement a basic user interface for chat.
* Store chat history in a text file.

**8. Temperature Converter**

**Description:** Develop a temperature converter that can convert temperatures between Celsius, Fahrenheit, and Kelvin.

**Features:**

* Convert temperature from Celsius to Fahrenheit and Kelvin.
* Convert temperature from Fahrenheit to Celsius and Kelvin.
* Convert temperature from Kelvin to Celsius and Fahrenheit.

**9. Number Guessing Game**

**Description:** Implement a number guessing game where the user has to guess a randomly generated number within a certain range.

**Features:**

* Generate a random number.
* Allow the user to input guesses.
* Provide feedback if the guess is too high or too low.
* Track the number of attempts.

**10. File Handling Project**

**Description:** Create an application to demonstrate basic file handling operations in Java.

**Features:**

* Create and write to a file.
* Read from a file.
* Append data to an existing file.
* Delete a file.