THINGS TO KNOW:

1. **Lab report must contain following sections: (order must be maintained)**
   a) **Title /Question**
   b) **Theory: The brief overview of the concept /techniques/syntax/technology used in the program**
   c) **Code: The complete code**
   d) **Output: Screenshot of the output**
2. **Output screen should be captured (use snipping tool), printed and attached in the report. Other contents must be handwritten.**
3. **Every source code must include the printing statements to print following information after your main output:**

   Lab No.:

   Name:

   Roll No./Section :

4. **Contents should be written on single side of A4 sized paper.**
5. **The works must be submitted within specified deadline.**
6. **Cover page and index page should be attached in the report appropriately.**

==========

Lab Works (Advanced Java Programming/CSIT-7th)

1. Create a class named "Box" which includes the variables: *length*, *breadth*, and *height*. Add a method to print the values of these variables. Add another method that prints the volume of box.
   Now create two objects of "Box" class inside main method and initialize the variables for both objects then call above methods using each object.

2. Create a class "Money" having two attributes "rupee" and "paisa" of type int and three methods: setMoney() - to set the values , displayMoney() – to display the values and addMoney(). The method addMoney() should have two parameters of type Money and it should compute the sum of those. Create another class "MoneyTest" having main method. Now create three objects of Money class inside main , initialize two of them using setMoney(), call addMoney() by passing those two objects and display the values of all three methods. [Result of addition should be like: 5 Rupee,75 Paisa + 7 Rupee ,30 Paisa = 13 Rupee 5 Paisa]

3. Perform the following tasks :
    a) WAP to create a single dimension array to store first 15 natural numbers and display the sum of 3rd and last element of the array.
    b) WAP to take the elements of a matrix of size 3*3 from user and display the matrix and all diagonal elements.
    c) WAP to demonstrate the concept of jagged array.
    d) WAP to demonstrate for-each loop in java.
4. WAP to illustrate:
    a) method overloading
    b) constructor overloading

5. WAP in Java to demonstrate the concept of
    a) static variable
    b) static method
    c) static block
    d) final keyword (variable/method/class)
6. Create a class **Student** with instance variable **roll_no** and two methods to read and display the roll no. Then, create another class **Test** that inherits class **Student**, consisting of its own instance variables to hold the marks of two subjects and also methods to read and display the marks. Finally, create another class **Result** which inherits class **Test**. It also has its own instance variable total that hold the total to hold the total of two marks scored by the student. Similarly, it has methods to calculate and display the total. Create some instances of above classes and demonstrate inheritance.

7. WAP to demonstrate
    a) the use of super keyword to access super class constructor.
    b) the use of super keyword to overcome name hiding (to access parent class members from child class)
    c) multilevel inheritance in java
    d) what happens when constructors are called in multilevel inheritance in java.
8. WAP to
    a) show method overriding in java
    b) illustrate runtime polymorphism in java (Hint: Area of Figures: Rectangle, Triangle)
    c) demonstrate Abstract class in java
    d) to demonstrate the use of interface to achieve multiple inheritance in java
9. WAP to demonstrate
    a) Static nested class
    b) Non static nested class (inner class)
    c) Local inner class
10. Write some java programs to show the techniques of importing packages in a java program. (Hint: fully qualified name, import packagename.classname; and import packagename.*;)
11. Create an interface called **Shape** which has methods **area (double x, double y)** and **perimeter (double x, double y)** to compute area and perimeter. Create a class called **Rectangle** which implements this interface. Declare instance variables as per requirement in class itself. Create instance of the class **Rectangle** and demonstrate interface implementation.
12. WAP
    a) to demonstrate try,catch and finally blocks in exception handling
    b) having multiple catch blocks in exception handling
    c) having nested try statements in exception handling
    d) to demonstrate the use of ‘throw’ keyword in exception handling
    e) to demonstrate the use of ‘throws’ keyword in exception handling
    f) to illustrate the idea of custom exceptions in java (user defined exceptions)

13. WAP
    a) to demonstrate the process of creating threads by implementing "Runnable" interface
    b) to demonstrate the process of creating threads by extending "Thread" class
    c) to show the use of isAlive() and join() methods of Thread
    d) to demonstrate the process of setting and getting thread priorities
    e) to illustrate the process of thread synchronization using synchronized method
    f) to illustrate the process of thread synchronization using synchronized block
    g) to create a thread that prints numbers from 100 to 1 in the interval of 3 seconds by extending the **Thread** Class.
    h) to create a thread that prints the numbers from 1 to 10 in the interval of 2 seconds by implementing the **Runnable** interface.
14. WAP
    a) to read data from file using **FileInputStream**
    b) to write data to a file using **FileOutputStream**
    c) to read from a image file and write to another image file
    d) to demonstrate the use **DataInputStream** class and **DataOutputStream** class
    e) to demonstrate the use of **RandomAccessFile** class for random accessing of file
    f) to show the use of **FileReader** class
    g) to show the use of **FileWriter** class
    h) to read the text from keyboard and write to a file
    i) to demonstrate the concept of serialization and deserialization in java