

# Object Oriented Programming using JAVA Lab

Subject Code :18CSI302L

Credits : 02

L-T-P:

0-0-4

## List of Experiments:

1. Demonstrate Constructor Overloading and Method Overloading in JAVA.
2. Implement Inner Classes and demonstrate its access protection.
3. Implement the following:
  - i) An abstract class "Shape" with the following properties: an instance variable shapeName of type String, an abstract method area (), a toString() method that returns the name of the shape.
  - ii) Create a subclass named "Sphere" which has radius and its area given by the formula  $4 \times \text{PI} \times r^2$ .
  - iii) Create a subclass named "Rectangle" which has length and width and its area is length times width.
  - iv) Create a subclass named "Triangle" which has base and height and its area is  $\frac{1}{2} \times \text{base} \times \text{height}$ .
  - v) Create another class which displays the calculated area.
4. Perform the following operations:
  - i) Check the length and capacity of String and StringBuffer objects
  - ii) Reverse the contents of a string given on console and convert the resultant string in Upper Case.
  - iii) Input a string from the console and append it to above resultant string.
  - iv) Extract the substring from resultant string.
5. Create the following:
  - i) A class "Account" with minimum balance 1000rs, deposit() method to deposit amount, withdraw() method to withdraw amount and also throws LessBalanceException if an account holder tries to withdraw money which makes the balance less than 1000rs.
  - ii) A class "LessBalanceException" which returns the statement that says "withdraw amount ( \_\_ rs) is not valid".
  - iii) A class which creates 2 accounts through which both deposit and withdraw operations are performed. Appropriate action has to be taken for LessBalanceException.

6. Implement Linear Queue using user defined exception handling (also use 'throw' and 'throws' keyword)
7. Implement the concept of Producer Consumer using synchronized threads.
8. Create the following:
  - i) Create an Interface for 'Stack' operations.
  - ii) A class that implements the Stack interface and create a fixed length stack.
  - iii) A class that implements the Stack interface and create a dynamic length stack.
  - iv) A class that uses the above stacks through interface reference and does the stack operations that demonstrates the runtime binding
9. Develop the following:
  - i) Create a package named "Calculator".
  - ii) Create some classes in the package representing some common operations like addition, subtraction, multiplication and division.
  - iii) Import and compile these classes in other program.
10. Create an enumeration DayofWeek with seven values SUNDAY through SATURDAY. Add a method isWorkday() to the DayofWeek class that returns true if the value on which it is called is MONDAY through FRIDAY.
11. Using File I/O streams, write a program to demonstrate file operations.
12. Write a Swing Application which uses:
  - i) JTabbed Pane
  - ii) Each Tab should use JPanel, which includes any one component given
    - i. below in each Panel
    - ii. ComboBox / List / Tree / Radiobutton