**M. S. Ramaiah Institute of Technology**

**Department of Information Science & Engineering**

**Subject: OOP Using Java Lab Subject Code: ISL38**

**Term: 28 Aug to 20 Dec 2018 Semester: 3**

**PART B**

1. Write Java programs
   1. To print fibonacci series without using recursion.(concept of loops, data types)
   2. To check prime numbers.
2. Create a class called account with the data members(Accno – integer, name String, Phone\_No: integer, balance\_amt:float), and following methods :
   1. getinput() to get input from the user
   2. Deposit() method which takes the amount to be deposited in to his/her account and do the calculation.
   3. Withdraw() method which gets the amount to be withdrawn from his/her account.
   4. Print the appropriate results.
3. Define a Stack class to implement the stack data structure. Include constructors to perform initialization, method push to push an element into the stack, method pop to remove an element from the stack and display method to display the elements of the stack.
4. Define a class Complex with data members as two real numbers, constructors for initialization these numbers, methods to add, subtract and multiply 2 complex numbers.
5. Write a java program to work with strings.
6. Extract a portion of the string and print it. Variable m indicates the amount of characters to be extracted from the string starting from the nth position.
7. Read a text and count all the occurrences of a particular word.
8. Replace a substring in the given string.
9. Rearrange the string and rewrite in alphabetical order.
10. Compare two strings ignoring case.
11. Concatenate two strings.
12. Create a Personal class to hold the personal details of an person such as name, age, education, salary-(0basic, da, hra), years of experience, number of loans and loan amount. Write constructors to assign values to the data members. Include
13. isEligible() method to indicate whether the person is eligible for loan,
14. taxPay() method to indicate the amount of tax to be paid,
15. isEligiblePromotion() to indicate whether the person is eligible for a promotion.
16. Display () method to display the details.
17. Enter the details of n employees and indicate their eligibility and the tax to be paid.
18. Write a Java Program that does the following related to Inheritance:
19. Create an abstract class called Vehicle which contains the ‘year\_of\_manufacture’ data member and two abstract methods ‘getData()’ and ‘putData()’ with a constructor.
20. Create two derived classes “TwoWheeler” and “FourWheeler” and implement the abstract methods. Make “FourWheeler” as final class.
21. Create class ‘MyTwoWheeler’ which is a sub-class of “TwoWheeler” and demonstrate the use of super keyword to initialize data members of “MyTwoWheeler”.
22. Write a Java Program that does the following related to Inheritance where:
23. Abstract class ‘Shirt’ with members size, price and color, constructors and abstract methods getColor() and putColor().
24. Inherit the Shirt class to obtain 2 classes: a final class “PullOver” (with members has\_hood, has\_stripes) and class “FormalShirt”( with members has\_full\_sleeve, has\_stripes).
25. Create class ‘PartyWear’ (with member brand) which is a sub-class of “FormalShirt” and demonstrate the use of super keyword to initialize data members of “PartyWear”.
26. Use packages and inheritance.
27. Define an interface ‘Department’ with methods to readdata() and printdata(), print\_number\_designations(), number\_research\_consultancy\_projs(). Define a ‘Faculty’ class with members name, designation, age, years of experience, joining\_date and subjects\_handled.
28. In package ISE define the ‘ISE\_department’ class that implements the ‘Department’ interface, accepts n faculty details and define all the methods. Raise a user defined exception ‘AgeException’ if the age of the faculty is > 58.
29. In the default package define a ‘MainClass’ which uses the methods of the above classes and also displays those faculty details whose years of experience is greater than or equal to 20.
30. Write a Java Program that does the following related to Packages and Interfaces , Exception Handling:
31. Create an interface ‘Student’ which gets the name and branch of a student.
32. Create a package called ‘StudentPackage’ which has a user-defined class RegisterStudent.
33. If a student registers above 30 credits for the semester, the method should throw a user-defined exception called ‘*CreditLimit*’ and display an appropriate message.
34. Create another package called ‘ResultPackage’ . Define a class ‘Grade’ to accept the grade for the 3 subjects for particular semester and compute the SGPA . If SGPA is above 10 then throw an ‘InvalidSGPA’ user-defined exception.
35. Write a class ‘Result’ to accept student details and compute the SGPA and raise appropriate exceptions.
36. Create an ‘Employee’ class with members name, age, qualification, and year of experience. Write a java program to create a LinkedList of Employee class. Only if the year of experience of an employee is greater than 5 display the details.
37. Write a java program to help the shop owner to create a Map of his customer’s id with the credit points. The customer id’s are in the range 100 to 200. For customer’s id from 105 to 120 update the credit points by 20. For customer’s id from 125 to 200 update the credit points by 30. Display the customer map.