**Lab Assignments** Subject: Core java

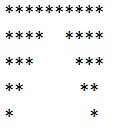
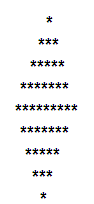
1. **Fundamentals**
2. **Using If-else ladder**

Implement a program to display the grading system of any students as per the following condition:

1. if per is greater than 70 then Grade will be A+
2. if per is greater than 60 & Less than 70 then Grade will be A
3. if per is greater than 50 & Less than 60 then grade will be B
4. if per is greater than 40 & Less than 50 then grade will be C
5. if per is less than 40 then remark as fail
6. **While, do-While loop**
7. Write a program to read a sequence of positive integers input by the user, and it will print out the average of those integers. The user is prompted to enter one integer at a time. The user must enter a 0 to mark the end of the data. (The zero is not counted as part of the data to be averaged.) The program does not check whether the user's input is positive, so it will actually add up both positive and negative input values.
8. Do same by using do-while

1. **For Loop**

Implement a program that prints the following pattern.

1

1. **Switch Statement**

Implement a menu driven program for switch statementwhere menus are as follows

1. Factorial

2. Fibonacci series

3. Sum of digits

4. Invalid choice

Make a Bank application using above fundamentals of Java having following operations.

1. Deposit money; 2. Withdraw money; 3. Check balance.

**5. Implement a program for following output (use break and continue)**

12346789

**2)**  **Number Handling**

1. **Even or Odd**

Implement a program to check given number is Even or odd.

1. **Magic Number**

Implement a program to check given number is Magic or not?

[Magic number=19=1+9=10=1+0=1]

226=2+2+6=10=1+0=1

1234=1+2+3+4=10=1+0=1]

1. **Prime number**

Implement a program to find the given number is Prime or not?

1. **Reverse number**

Implement a program to reverse the given number.

1. **Armstrong number**

Implement a program to check given number is Armstrong or not?

[Armstrong number=153=13+53+33=153]

1. **OOPS-I**
2. **Method Overloading**

Implement a program to add 5 numbers using method Overloading.

1. **Method overriding**

Write a ‘swift’ class with base class as ‘car’ to calculate a speed of car by using method overriding concept.

1. **Constructor**

Write an Employee class which having the data members as emp\_id, emp\_name, emp\_desig and methods used in it are emp\_getadata(),emp\_setdata() and emp\_claSal() make use of constructor.

1. **Constructor Types**

Implement a cube class using parameterized constructors.

1. **Constructor overloading**

Implement a StudentInfo class having the data members as Stud\_Id,Stud\_Name,andStud\_Age and the methods are getdata() and setdata() by using constructor overloading.

1. **OOPS-II**
2. **Copy constructor**

Implement a program to copy the values of one object into another object using demo class with the help of copy constructor.

1. **Static variable and static method**

Write a class Stat\_demo to illustrate a use of static variable and static method.

1. **This keyword**

Write a class to demonstrate a use of this keyword.

1. **Super keyword**

Write a class to demonstrate a use of Super keyword.

1. **Final keyword**

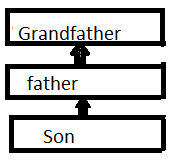
Write a class to demonstrate a use of final keyword.

1. **OOPS-III**
2. **Simple Inheritance**

Write a vehicle and two\_whlr class by using proper variables and methods for it.

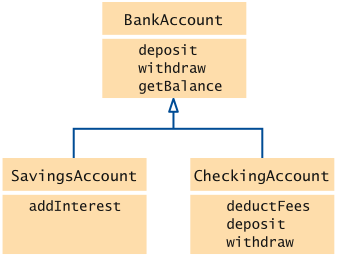
1. **Multilevel Inheritance**

Implement the following

****

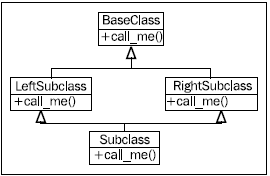
1. **Hierarchical Inheritance**

Implement the following



1. **Hybrid Inheritance**

Implement the following

****

1. **Final method and class**

Implement a simple program which uses Final method and Final class.

1. **Abstract Class and Interface**

Implement a program to illustrate a use of abstract class and interface.

1. **String Handling**
2. **Comparison**

Implement a program for string comparison by two methods

i. compareTo ()

ii.equals()

1. **Occurrence**

Implement a program to count number of occurrence of ‘n’ in a given string

“Hello Hunny Bunny”.

1. **Reverse**

Implement a program to display a reverse string with and without reverse ().

1. **Append**

Implement a program to append a character in any string (take input from user).

1. **Merge**

Implement a program for following output

i/p=s1: {‘1’,’3’,’5’,’7’} and s2={‘2’,’,4’,’6’,’8’}

o/p=s3={‘1’,’2’,’3’,’4’,’5’,’6’,’7’,’8’}

1. **Exception Handling**
2. **Arithmetic Exception**

Implement a simple program to handle arithmetic Exception.

1. **Multiple Try-Catch Blocks**

Implement a program to illustrate a use of multiple try-catch blocks**.**

1. **Throw keyword**

Implement a program to illustrate a use of Throw keyword.

1. **Throws keyword**

Implement a program to illustrate a use of Throws keyword.

1. **Finally keyword**

Implement a program to illustrate a use of finally keyword.

1. **Multithreading**
2. **Extending Thread**

Implement a simple program to extend the thread.

1. **Sleep()**

Implement a program to demonstrate the use of Sleep ().

1. **Join()**

Implement a program to demonstrate the use of Join ().

1. **Priority thread**

Implement a program which uses priority thread.

1. **Thread pool**

Implement a program to demonstrate the use of Thread pool.

1. **File Handling**
2. **Character Stream**

Implement a program to read and write into a file using character stream.

1. **Byte Stream**

Implement a program to read and write into a file using Byte stream.

1. **Copying of data**

Implement a program to copy data from one file to another file use any stream.

1. **Buffered Stream**

Implement a program to read and write into a file using Buffered stream.

1. **Count no of Characters**

Implement a program to count no of characters from a given file, and use any one stream.

1. **Serialization**
2. **Serialization**

Implement a program for serialization.

1. **Deserialization**

Implement a program for Deserialization.

1. **Serialization with inheritance**

Implement a program for serialization with inheritance.

1. **Stream handler**

Implement a program for Stream handler**.**

1. **Transient keyword**

Implement a program to demonstrate the transient keyword.

1. **Networking**
2. **Socket programming**

Implement a program for socket programming.

1. **URL Classs**

Implement a program for URL class.

1. **InetAddress Class**

Implement a program for InetAddress Class.

1. **DatagramSocket and datagrampacket**

Implement a program for DatagramSocket and datagrampacket

1. **Displaying Web page content**

Implement a program to diasplay the content of Web page.

1. **AWT Event Handling**
2. **Button**

Implement a simple program for event handling and add ‘ok’ button into window.

1. **MenuItem**

**Implement a menu driven program for the menus file,Home,insert,Font by using appropriate Events.**

1. **CheckBox,Radio Button**

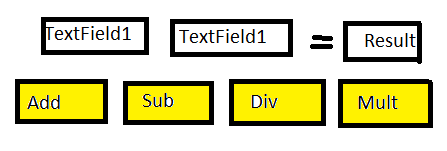
**Implement a program for following output**

1. **Select city of choic 2. Select Gender**

****

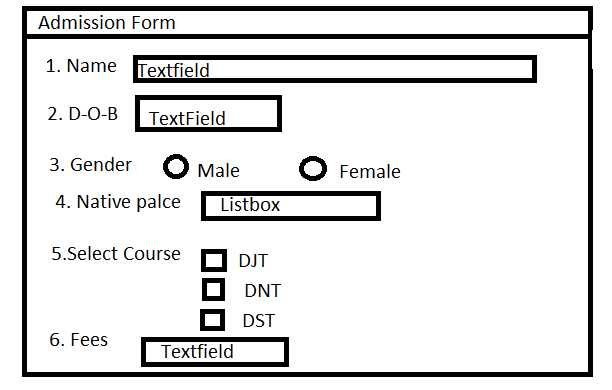
1. **Text field**

Implement a program for the following figure.

****

1. **Making of Form**

Implement a program for following output.

****

1. **Swing**
2. **JComboBox class**

Implement a program to demonstrate a use of JComboBox class.

1. **Display Image**

Implement a program to display the image by using swing.

1. **Graphics in Swing**

Implement a program to draw circle,oval,rectangle,square and Triangel using swing.

1. **JColorchooser class**

Implement a program to illustrate a use JColorchooser class

1. **JTextField Class**

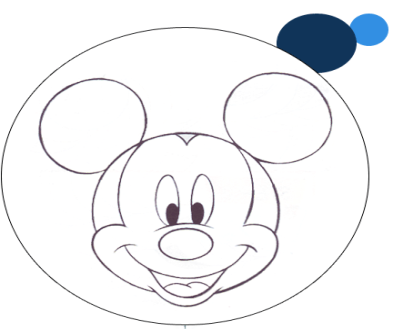
Implement a program to illustrate a use JTextField Class

1. **Applet**
2. **Creation of Applet Window**

Implement a program to create a simple window of applet with close, minimize and maximize option of window.

1. **Applet Graphics**

Draw following figure by using applet.[it seems to be like this]

****

1. **Animation in Applet**

Implement a program to demonstrate a use of animation in applet.

1. **Event handling in Applet**

Implement a program to make Railway reservation form with the help of Event handlings in Applet.

1. **Display image**

***Implement a program to display a image in applet Window***

1. **Collection, Generics**
2. **Arraylist Class**

Implement a program to demonstrate a use of ArrayList class

1. [**ListIterator interface**](http://www.javatpoint.com/ListIterator-interface-in-collection-framework)

Implement a program to demonstrate a use of [ListIterator interface](http://www.javatpoint.com/ListIterator-interface-in-collection-framework).

1. **HashSet Class**

Implement a program to demonstrate a use of HashSet class.

1. **TreeSet Class**

Implement a program to demonstrate a use of TreeSet Class.

1. **Sorting Class**

Implement a program to demonstrate a useof Sorting Class.

1. **JDBC**
2. **JDBC Connection**

Implement a program to make a connection with JDBC Drivers.

1. **Connection with MYSQL**

Implement a program to make a connection with JDBC Drivers by using MYSQL database.

1. **Connection with Oracle**

Implement a program to make a connection with JDBC Drivers by using Oracle database.

1. **Retrieve Image**

Implement a program to Retrieve Image.

1. **Transaction management**

Implement a program for Transient management.