

## JAVA I

### Lab Assignment-6

1. Create a class Book with **member variables** book\_id and pages. Then create a subclass FictionBooks with **member variable** called name. Create some instances of FictionBooks class and set the values of both subclass and superclass and display all member variables that have been initialized.
2. Create a class Publication which has instance variables title and price. From this class derive two classes Book and CDROM. Book class has instance variable page and CDROM class has instance variable playtime. Both these classes have getdata () method to get input from the user and putdata () to display the contents. Write a program to create instances of these classes and demonstrate appropriate operations.
3. Create a class called Square with two methods area () and perimeter () that returns double values. Create another class Rectangle that is a subclass of class Square. **Override** the definitions of both area () and perimeter () methods in Rectangle.
4. Write a program according to the following specifications:
  - Create a class Student with instance variables roll no and two member methods to read and display the roll no.
  - Create a class Test that inherits class Student. It consists of its own instance variables to hold the marks of two subjects and also member methods to read and display the marks.
  - Create another class Result which inherits class Test. It has its own instance variable total to hold the total of two marks scored by the student. Similarly, it also has member methods to calculate and display the total.
5. Create a class Account that has instance variables customer\_name, account\_number, account\_type and balance. From this class derive class SavingAccount and make it more specific to need. Include required member methods to do the following:
  - Accept the deposit from the user and update the account
  - Display the balance
  - Update the account after withdraws.
6. Create a class Lecturer with instance variables id and name. From Lecturer class derive two classes Part\_Time, which has instance variable payperhour and Full\_Time, which has instance variable paypermonth. All these classes have getdata() method to get input from the user and putdata() to display the result. [Make the getdata() and putdata() methods of Lecturer class as abstract]. Create some instances of Part\_Time and Full\_Time classes and show the necessary operations.
7. WAP to demonstrate that a **superclass** variable can hold the reference of a **subclass** object.
8. WAP to demonstrate the concept of **Dynamic Method Dispatch** to overcome method overriding.

9. Create a class called Maths with instance variables num1 and num2 and member method input() to get input from the user. Derive the following classes from Maths class:
  - i) Add (Create appropriate method to add the two numbers of Maths class and display the result)
  - ii) Sub (Create appropriate method to subtract the two numbers of Maths class and display the result)
  - iii) Mul (Create appropriate method to multiply the two numbers of Maths class and display the result)
  - iv) Div (Create appropriate method to divide the two numbers of Maths class and display the result)
  - v) Rem (Create appropriate method to find remainder obtained while dividing the two numbers of Maths class and display the result)
10. Create an abstract class called Fmachine. Derive a classe Airplane, having instance variables code, name, capacity and methods getdata() and putdata() to read and display the result. Create some instances of Airplane and call the required methods.
11. Create an abstract class College with instance variable college\_id and college\_name and member methods get() and put(). Create another class Student that derives class College which has instance variables roll, name and address and methods get() and put(). WAP to create some instances of Student class and demonstrate the methods of College class being overridden.
12. Create an interface Shape which has methods get\_data() and display\_area(). Create two classes Rectangle and Triangle which implements this interface (Define the instance variables of these classes as per the requirement). Create some instances of Rectangle and Triangle classes and demonstrate interface implementation by classes.
13. Create an interface College with instance variable college\_id and college\_name (college\_id and college\_name should be fixed by the user) and member methods get() and put(). Create another class Student that derives class College which has instance variables roll, name and address and methods get() and put(). WAP to create some instances of Student class and demonstrate the methods of College class being overridden.
14. Create two interfaces Pulication and Sale. Pulication has instance variables title and price and member methods getdata() and display(). Similarly, Sale has instance variable publishesales and member methods getdata() and display(). Create a class Book which implements both the interfaces. It has got instance variable pages and member methods getdata() and display(). Write a program to create some instances of the Book class and demonstrate the implementation of interface in the class.
15. WAP to demonstrate the uses of **super** keyword.
16. WAP to demonstrate the order of **constructor** invocation.
17. Demonstrate the different uses of **final** keyword.
  - To declare a variable as constant

- To prevent overriding
- To prevent inheritance

18. Write a program to demonstrate the use of Interface extending another interface.
19. Write a program to demonstrate the concept of nested interface.
20. Write a program to demonstrate the concept of partial interface implementation.
21. Create a package named mypackage and define a class ArithmeticOperations within this Package with the following methods:

```
int sub (int, int) { }  
int multiply (int, int) { }  
int division (int, int) { }  
int factorial (int) { }  
int square (int, int) { }
```

(Save this package inside the subdirectory named my package in root directory or while compiling compiler also makes a sub-directory named mypackage (i.e. package name)). Give the name of the file Arithmeticoperations.java. Now write a program to use this package, use `import mypackage`)