(An Integrated Lab with [illegible] – OOP using JAVA)

Tentative Programs

OBJECTIVE:

- Design simple java programs using classes, objects, inheritance and polymorphism.
- Experience how inheritance and the code reusability feature works.
- Design and implementation of java programs using Constructors, overloading, type conversion and String handling.
- Design and implementation of programs using Threads and Exception Handling.
- Design and implementation of Simple Applets and experience the working of applets.
- Design and implementation of programs using Java I/O streams.

CYCLE-II

Develop and Execute applications for the following:

1. Develop an application with an abstract Bank class consisting of below features:

    a) Bank_Type(National/International)

    b) getRoi() is an abstract method.

    c) Deposit().

    d) OfferCreditCard.

    - Identify various (atleast 4) banks which will extend bank class with below conditions:

    - If the bank is type of nationalized the roi is 7% else 8% . The interest is offered only if the term deposit is minimum of 3 years.

    - If the bank is international then it can offer platinum credit card otherwise not.

    - Override one of the method and use the concept of dynamic method dispatch.

2. class Student consists of below attributes:

    a) Rollnumber

- class Test inherits Student and will identify marks for 2 subjects for each student and will display the same.
- Each student can also be recognized with sports weightage using an interface Sports which will have attributes to assign weightage value and display it. Each Student is rated with total score which is the summation of marks in 2 subjects and sports weightage.
- Develop an application Results to extend Test and implement Sports which displays the student score card as below:

  Roll No:

  Marks Obtained

  Subject1 and Subject2

  Sports weight:

  Total Score

3. Develop an application which consists of Stack class and include methods to push and pop the stack elements. Create your own defined Exception class and handle exceptions while stacks run with “Stack as full” and “Stack under flow” while calling push and pop methods respectively. Implement this using throws and throw keywords.