# Academy of Technology

## Department of Computer Science & Engineering, Semester: 5TH

## *Paper Name: Object Oriented Programming Lab*

## *Paper Code: PCC-CS593*

## Laboratory Assignment: 7

**7.1** Write a program by using class with the following specifications:
Class name — Sale
Data members/ Instance variables:

1. String title, author, publication
2. double price

Member methods:

1. void input() — to accept title, author name and publication name and price of a book
2. void display() — to display title, author name and publication name and price of a book

Now, create another class 'Purchase' that inherits class 'Sale' having the following specifications:
Class name — Purchase
Data members/ Instance variables:

1. int noc
2. int amount;

Member methods:

1. void accept() — to enter the number of copies purchased
2. void calculate( ) — to find the amount by multiplying number of copies ordered and price (i.e., noc * price)

3. void show() — to display the elements describes in base class along with the number of copies purchased and amount to be paid to the shopkeeper

**7.2** Write a program to use a class Account with the following specifications:
Class name — Account
Data members — int acno, float balance
Member Methods:

1. Account (int a, int b) — to initialize acno = a, balance = b
2. void withdraw(int w) — to maintain the balance with withdrawal (balance - w)
3. void deposit(int d) — to maintain the balance with the deposit (balance + d)

Use another class Calculate which inherits from class Account with the following specifications:
Data members — int r,t ; float si,amt;
Member Methods:

1. void accept(int x, int y) — to initialize r=x,t=y,amt=0
2. void compute() — to find simple interest and amount
   si = (balance * r * t) / 100;
   a = a + si;
3. void display() — to print account number, balance, interest and amount

**7.3** Write java programs to illustrate Single Inheritance, Multilevel Inheritance. Also, write java programs to explain super & method overriding.

**7.4** Write a java program to implement runtime polymorphism.