# 03. OOPs – Assignment

1. Create a class Tree which specifies the height of the tree. The default constructor should print “Planting a tree”.
2. Write a program to count the numbers of instances created for a class.
3. Create a class for a Product with productno (int), productname (String), price (float) and quantity (int) as attributes. Create a method to initialize variables (attributes) and another method to invoke values of attributes. Finally print the output in console in following format.

Product No Prduct Name Quantity Price Total Amount

Value value value value value

1. Create a class Account with the attributes accountNo – int, accountType – String, accountBalance – int and methods setAccountDetails() which set the values to these attributes, withdraw() which subtracts 1000 from the available balance, deposit() which adds 1000 to the available balance and dispAccountDetails() which displays accountNo, accountType,accountBalance. Create a class Main1 which contains main() method to test the functionality of Account class.
2. Create a class Student with the attributes rollNo – int , name - String, mark1-int,mark2-int,mark3-int,total – int and methods setStudDetails() which set the values to these attributes except total, findTotal() which is used to find the total marks and dispStudDetails() which displays rollNo, name and total. Create a class Main2 which contains main() method to test the functionality of Student class.
3. Create a Bank class with methods deposit & withdraw. The deposit method would accept attributes amount & balance and returns the new balance which is the sum of amount and balance. Similarly, the withdraw method would accept the attributes amount & balance and returns the new balance 'balance-amount' if balance >= amount or return 0 otherwise.
4. Create Rectangle class which provides information on length and breadth. Create two parameterized constructors. First constructor has only one parameter which initializes length and breadth with same value. Second constructor has 2 parameters which initializes length and breadth with different values. Use this to initialize values for the attributes in second constructor.
5. Create an abstract class called Employee with following attributes and methods.

Empid(int), empname(String) -> Attributes

input(int,String) -> Concrete Method

output() -> Concrete Method

roles\_Responsibilities() -> abstract method

Create a class called Manager which is a subclass of Employee and override the abstract method of Employee. Invoke all the methods.

1. Make a class Employee with a name & salary. Make a class Manager inherit from Employee. Add an instance variable, named department of type string. Supply a method toString() that prints the manager's name, department and salary. Make a class Executive inherit from Manager. Supply a method toSting() that prints the string “Executive”, followed by name, department and salary defined in the Manager superclass. Tests these classes and methods from main method.
2. Create an Interface called Calculator. Implement it and access the services for addition and subtraction from an application.