

Exercise 1

```
interface IDisplay {
    void print(); // Print in one line
    void printDetails(); // Print in multiple Lines
}

interface IInput {
    void input();
}
```

- a) Implement a class called Book with properties bookID, title, publisher which uses the interfaces IDisplay and IInput
- b) Implement a class called Student with properties studentID, and name which uses the interfaces IDisplay and IInput
- c) Create objects of the Book and Student in the main method.
- d) Create a variable of the IDisplay and IInput interfaces and call the printDetails() and input() methods respectively of Book and Student objects.

Exercise 2**Account Class**

- a) Implement an abstract class called Account. Have the following properties accountNo, name, balance.
- b) Implement a Deposit() method to deposit money. The amount deposited should update the balance.
- c) Implement a constructor to get values to the Account class.
- d) Have an abstract method called calculateInterest() which returns a double value.
- e) Implement a method to display() the account details.

FixedDepositAccount Class

- f) Implement a new class called `FixedDepositAccount` which extends the `Account` class.
- g) It should have a new property called `interestRate` and `Interest`.
- h) Write a setter and getter for the `interestRate`.
- i) Implement the `calculateInterest()` method assuming that the Balance has been held for the entire year. `interest = balance * interestRate/100;`

SavingsAccount Class

- j) Implement a new class called `SavingAccount` which inherits the `FixedDepositAccount` class
- k) Implement a `withdraw()` method that allows you to withdraw money from the `SavingsAccount`.
- l) Implement the `calculateInterest()` method assuming that the Balance has been held for the one Month. `Interest = balance * interestRate/100/12;`

Create objects from the `FixedDeposit` and `SavingAccount`. Call the `Deposit()` and `withdraw()` methods (Only `SavingsAccount` have withdrawals)