

Java OOP Assignment

1. Write a program which creates a class "Student" with the following

Data Members

rollNum, studName, mark1, mark2, mark3, totalMarks

Methods

setStudDetails() – which sets the values to all the data members except totalMarks.

calculateTotal() - which calculate the totalMarks

displayStudDetails() – which displays rollNum, studName and totalMarks

Create a class StudentDemo to test the functionality of Student class

2. 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 the given amount from the available balance, deposit() which adds given amount 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.
3. Convert the above two programs using constructors.
4. Write a program to create a class Rectangle with datamembers length and breadth. Create a method area() which finds the area of the rectangle. Use constructors to assign values to datamembers
5. Create a class Customer with the fields custId, custName and custAddress. Create a method display() which displays the Customer details.
Create a class Account with the fields acctId, acctType, Customer and acctBalance.
Create a method display() which displays Account Details
Create a main class "AccountTest" to test the above classes
Note: Use constructors to set the values to the fields of the classes
6. Write a program to find the number of instances created for a given class.

7. Create a class Book with the data members bno,bname,price.
Create a class SpecialEditionBook which is subclass of Book and with a data member discount. Create a class BookTest with main() to test the above classes

Note

- Use constructors to give the values to the data members
- create a method display() in Book class which displays bno,bname and price and override the same display() of Book class in SpecialEditionBook which displays bno,bname,price using super.display() and display the value of discount.

8. Create an abstract class "Fruit" which contains an abstract method cost().
Create sub classes "Apple and Mango" which extends Fruit class. Use appropriate data members in the respective classes. Create a class FruitDemo to test the above functionalities.

9. Modify the above program using interfaces

10. Create a package "com.account".

Create a class "Account" with the fields acctId, acctType, acctBalance. Use Constructor to set the values to these fields. Create the following methods in the Account class

```
void deposit(int amount)
void withdraw(int amount)
int getBalance()
void display()
```

Create a package "com.account.test".

Create a class "AccountMain" with the main() and displays the following menu

1. Deposit
2. Withdraw
3. Display Balance
4. Exit

Note:

The menu should display repeatedly until the user chooses Exit and the input need to accepted during runtime using Scanner class