# CSE 143x Section Handout #6

## Questions

### Objects

**3. BankAccount.** Create the header and the fields for a bank account class. Each BankAccount object should represent the user's name and the total balance of money.

**4. withdraw.** Add a method to the BankAccount class that takes in an integer representing the amount of money to be removed from the account's balance and updates the field accordingly.

**5. deposit.** Add a method to the BankAccount class that takes in an integer representing the amount of money to be added to the account's balance and updates the field accordingly.

**6. transactionFee.** Add a double field to the BankAccount class named transactionFee that represents an amount of money to deduct every time the user withdraws money. The default value is $0.00, but the client can change the value. Deduct the transaction fee money during every withdraw call (but not from deposits). Make sure that the balance cannot go negative during a withdrawal. If the withdrawal (amount plus transaction fee) would cause it to become negative, don't modify the balance value at all.

**10. euclideanDistance.** Write a method called euclideanDistance that takes two NDPoints and returns the euclidean distance between the two as a double. The euclidean distance is defined as the length of the line segment between both points, and can be calculated as the square root of the sum of the squares of the differences of each coordinate element. If the NDPoints passed have have a different amount of dimensions(e.g., their coordinates fields are of differing lengths), then you should throw an IllegalArgumentException). You can assume that the fields of both NDPoints are initialized.

```
public class NDPoint {
    double[] coordinates;
}
```

An example:

```
A(3.0, 4.0, 9001.1)
B(6.0, 7.0, 42.0)
distance = sqrt((3.0 - 6.0)^2 + (4.0 - 7.0)^2 + (9001.1 - 42.0)^2)
distance = 8959.10...
```