**Advanced Programming Tutorial & Lab**

Objective: The objective of this tutorial is to refresh your basic object oriented programming: Classes , Methods, Constructors and Inheritance.

1. Review Questions:

1. What are the parts of a class?
2. What is the relationship between a class and an object?
3. How long does the data stored in an object persist? How is it different from local variables?
4. What is the process for initializing the data in an object?
5. How are objects created in a program?
6. How are objects accessed?
7. Why direct access to object’s data must be restricted?
8. How do we access the data stored in an object?
9. How do we change the data stored in an object?
10. What is the purpose of the keyword this?

2. The following program was written to model a bank account that allows customers to withdraw and deposit funds. Assume no customer can withdraw more funds than what is in the current balance.

1. Identify all design and syntax errors in the program below (There are at least 5).
2. Modify the withdraw method so that it tracks the number of withdrawals done.
3. Add another method to add interest based on the current interest rate supplied.

class Account

{

private String name;

private String ID;

public double balance;

public void Account(String name, String ID, double balance)

{

name = name;

ID = ID;

balance = balance;

}

public void getBalance()

{

return balance;

}

public void withdraw(double amt)

{

balance -= amt;

}

public void deposit(double amt)

{

balance += amt;

}

}

public class TestAcct

{

public static void main(String args[])

{

Account a1 = new Account("Tan A K", "S123", 24.5);

Account a2 = new Account("Smith T","S124",1200.0);

a1.deposit(100);

a1. withdraw(2000);

a2.deposit(120);

a2.withdraw(80);

System.out.println("Balance for " + a1.name + " is " + a1.getBalance());

System.out.println("Balance for " + a2.name + " is " + a2.getBalance());

}

}

3. Extend the account class to create a new subclass called CAccount for modeling a checking account, with the following features:

* + Checking accounts can use an overdraft facility, so the new class will need instance variables for the overdraft limit and the amount overdrawn.
  + The overdraft should not be used unless the current balance is insufficient to cover a withdrawal.
  + When funds are deposited and there is an overdrawn amount, the funds should be used to reduce the overdrawn amount to 0 before the balance can be increased. For example, if overdrawn amount is $200, when $500 is deposited, overdrawn amount should be reset to 0, and the remaining amount $300 should be added to balance setting it at $300.
* A new constructor will be required that accepts the account name, id, initial balance and overdraft limit. This constructor should use the super() facility to initialize the account id, name and balance.
* This constructor should set the amount overdrawn to 0.
* Both withdraw and deposit methods must be overridden to accommodate these requirements. Test the class. Meet these requirements.

**Lab Exercise: The lab exercise in week 2 will be the diagnosis test**