

Example: Class Account

```
/**
 * A simple bank account for which the balance can never be
 * less than zero
 *
 * @invariant getBalance() >= 0
 * @invariant getId() is unique and set when account is created
 * @invariant getName() is set when account is created
 * @invariant the values of getId() and getName() never change
 */
public class Account
{
    private int id;
    private static int nextAccountId = 0;
    private String name;
    private double balance;
```

should the class have methods:

setId ?	YES	NO
setName ?	YES	NO

CPSC 211, Winter 2008, Term 1

1

```
/**
 * Initializes an account
 * @param accountName Customer name for account
 * @param initialBalance Initial balance deposited in account
 *
 * @pre true
 * @post getName() = accountName
 * @post getId() = a new number not returned by other accounts
 * @post (initialBalance >= 0 AND getBalance() = initialBalance)
 *       OR getBalance() = 0
 *
 */
public Account(String accountName, double initialBalance) {
    id = nextAccountId++;
    name = accountName;
    if (initialBalance >= 0)
        balance = initialBalance;
    else
        balance = 0;
}
```

CPSC 211, Winter 2008, Term 1

2

```

/**
 * Accessor method to return the account id
 * @pre true
 * @return the account id
 */
public int getId() {
    return id;
}

/**
 * Accessor method to return the customer name
 * @pre true
 * @return the customer name
 */
public String getName() {
    return name;
}

```

```

/**
 * Deposit money into the account
 * @param amount The amount to be deposited
 *
 * @pre amount >= 0
 * @post getBalance() = @pre.getBalance() + amount
 * @return The current balance of the account
 */
public double deposit(double amount)
{
    assert amount >= 0;

    balance = balance + amount;
    return balance;
}

```

```

/**
 * Withdraw money from the account
 * @param amount The amount to be withdrawn
 *
 * @pre amount >= 0
 * @post IF (@pre.getBalance()-amount >= 0 )
 *       THEN getBalance() = @pre.getBalance() - amount
 *       ELSE getBalance() = @pre.getBalance()
 * @return The current balance of the account
 */
public double withdraw(double amount) {

    assert amount >= 0;

    if (balance - amount >= 0)
        balance = balance - amount;

    return balance;
}

```

```

/**
 * Returns the string representation of an account
 *
 * @pre true
 * @return the account represented as a string
 */
public String toString()
{
    return "[ id = " + id + ", name = " + name +
           ", balance = " + balance + " ]";
}
}

```