



SOFTWARE TESTING

THE ROLE OF A TESTER IS TO BREAK THE SOFTWARE

Prepared For :
SIR MUKHTAIR ZAMIN

Prepared By :
BASIT IQBAL
FA21-BSE-050

Question:

Implement the Error Check List?

1. ALL Variables declared?
2. Default attributes understood?
3. Arrays and strings initialized properly?
4. Correct lengths, types and storage classes assigned?
5. Initialization consistent with storage class?
6. Any variables with similar names?

implement a scenario for this list

Answer:

Scenario:

Simple Bank Account Example:

Incorrect Code:

```
package ErrorCheckList;

/**
 *
 * @author BASIT
 */
public class BankAccount {

    private String name;
    public int balance;

    public BankAccount(int startingAmount) {
        amount = startingAmount;
    }

    public void deposit(double depositAmount) {
        balance += depositAmount;
    }

    public void withdraw(int withdrawalAmount) {
        if (withdrawalAmount > balance) {
            System.out.println("Insufficient Funds!");
        } else {
            balance = balance - withdrawalAmount;
        }
    }
}
```

Correct Code:

```
package ErrorCheckList;

public class CorrectBankAccount {
    private String accountHolderName;
    private double accountBalance;

    public String getAccountHolderName() {
        return accountHolderName;
    }

    public void setAccountHolderName(String accountHolderName) {
        this.accountHolderName = accountHolderName;
    }

    public double getAccountBalance() {
        return accountBalance;
    }

    public void setAccountBalance(double accountBalance) {
        this.accountBalance = accountBalance;
    }

    public void deposit(double depositAmount) {
        if (depositAmount > 0) {
            setAccountBalance(getAccountBalance() + depositAmount);
        } else {
            System.out.println("Invalid deposit amount. Please enter a positive value.");
        }
    }

    public void withdraw(double withdrawalAmount) {
        if (withdrawalAmount > 0 && withdrawalAmount <= this.accountBalance) {
            setAccountBalance(getAccountBalance() - withdrawalAmount);
        } else {
            System.out.println("Insufficient funds. Withdrawal amount cannot be negative or exceed your ba");
        }
    }

    public void displayAccountDetails() {
        System.out.println("Account Holder Name: " + this.accountHolderName);
        System.out.println("Account Balance: $" + this.accountBalance);
    }
}
```

All Variables Declared	
Incorrect Version	Correct Version
<pre> public class BankAccount { private String name; public int balance; public BankAccount(int startingAmount) { amount = startingAmount; } </pre>	<pre> public class CorrectBankAccount { private String accountHolderName; private double accountBalance; public CorrectBankAccount(double startingAmount){ accountBalance = startingAmount; } </pre>
The variable “amount” that is being used in the constructor is not declared.	In this version of code all the variables are declared.

Default attributes are understood	
Incorrect Version	Correct Version
<pre> public class BankAccount { private String name; public int balance; public BankAccount(int startingAmount) { amount = startingAmount; } </pre>	<pre> public class CorrectBankAccount { private String accountHolderName; private double accountBalance; public CorrectBankAccount(double startingAmount){ accountBalance = startingAmount; } </pre>
The variable “balance” is declared with public int and its default value is zero (0) which might not be intended.	The variable “accountBalance” is declared in this version of code whose initial value is initialized when its object is created.

Incorrect Type/Length Assignment	
Incorrect Version	Correct Version
<pre> public void deposit(int depositAmount) { balance += depositAmount; } </pre>	<pre> public void deposit(double depositAmount) { if (depositAmount > 0) { setAccountBalance(getAccountBalance() + depositAmount); } else { System.out.println("Invalid deposit amount. Please enter a positive value."); } } </pre>
The deposit Amount is of the type int while the balance is of the double type. This may lead to data loss.	In this version of code the deposit Amount is of correct type (i.e double)

Inconsistent initialization	
Incorrect Version	Correct Version
<pre> public class BankAccount { private String name; public int balance; public BankAccount(int startingAmount) { amount = startingAmount; } } </pre>	<pre> public class CorrectBankAccount { private String accountHolderName; private double accountBalance; public CorrectBankAccount(double startingAmount){ accountBalance = startingAmount; } } </pre>
The “balance” is not being initialized in the constructor, relying on default value.	In this version of code the “accountBalance” is being initialized in the constructor.

Similar names of variables	
Incorrect Version	Correct Version
<pre> public class BankAccount { private String name; public int balance; public BankAccount(int startingAmount) { amount = startingAmount; } } </pre>	<pre> public class CorrectBankAccount { private String accountHolderName; private double accountBalance; public CorrectBankAccount(double startingAmount){ accountBalance = startingAmount; } } </pre>
The “balance” & “amount” can be confused.	In this version of code the “accountBalance” is being used and amount is removed.

Test Cases:

Test Case Id	Test Case Description	Test Input Data	Expected Output	Actual Output	Verdict
TC_001	The amount will be stored in the declared variable.	account Balance=10,250	Amount saved.		
TC_002	The amount can be stored even if it is in decimal value.	account Balance = 100.12	Amount saved.		
TC_003	To check that the amount entered is added successfully into the current balance without data loss due to incorrect type.	deposit Amount=10.12	Amount added to current balance successfully		
TC_004	To check that at the creation of an account there is no variable, which is relying on the default value.	account Balance=0	Account Created Successfully		
TC_005	To Check that there is no confusion between the variable names.	account Balance=100 amount =10	Current Balance shows the balance in account while the amount shows the amount to be added.		