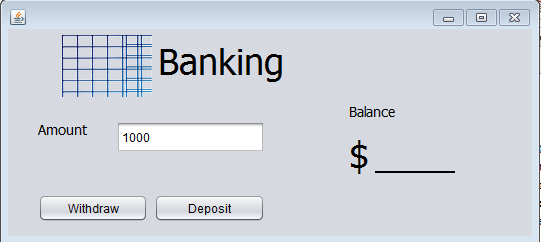
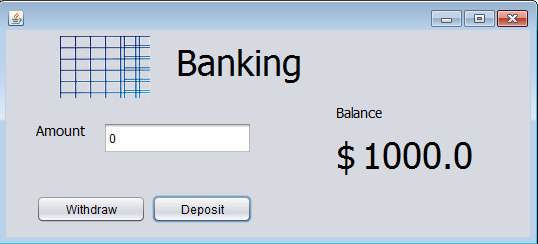
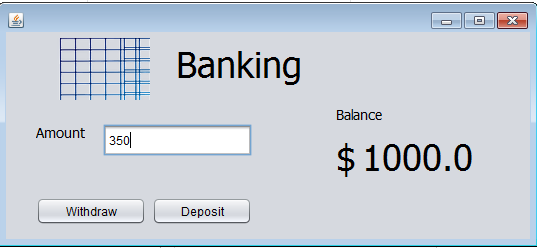
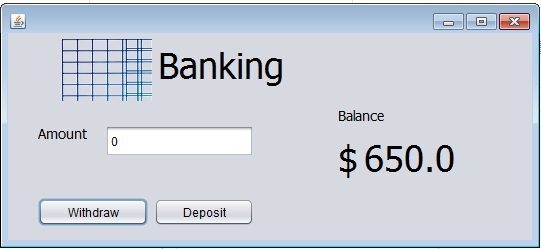
## Objectives:

* Use of Classes
* User of Methods
* Use of GUI (Graphical User Interface)
* Use of class inharitance

## Scenarios:

* 1. Expanding on the bank account class, create a graphical user interface that will allow the user to either deposit or withdraw money from the account and display the balance. You need to create your GUI. Below is a n idea of how the solution should look like:
* Open the solution to deposit $1000. Enter 1000 into the text field, then click deposit



* Notice the balance change
* 
* Withdraw $350
* 
* Notice the balance change
* 
  1. Modify the BankAccount class previously created to add two methods: deposit and withdraw. These will be used to deposit of withdraw money from the bank account. Extend the BankAccount class to create two types of bank accounts: a checking account and a savings account. These will inherit the properties of bank account class, but have the following unique behavior:
     + When money get deposited into the savings account, an automatic 1% interest is added to the deposited amount
     + The savings account does not allow withdraw
     + When money get withdrawn from the checking account, a withdrawing fee of $1 is applied.

To test the functionalities of the BankAccount, CheckingAccount and SavingsAccount classes a class called BankTest will be created. This class will have a main method which will be used to:

* Create at least one instance of each bank account type, i.e BankAccount, CheckingAccount and SavingsAccount
* Deposit/withdraw money from these accounts
* Display (output) the balance and interest values from these accounts

## Grading:

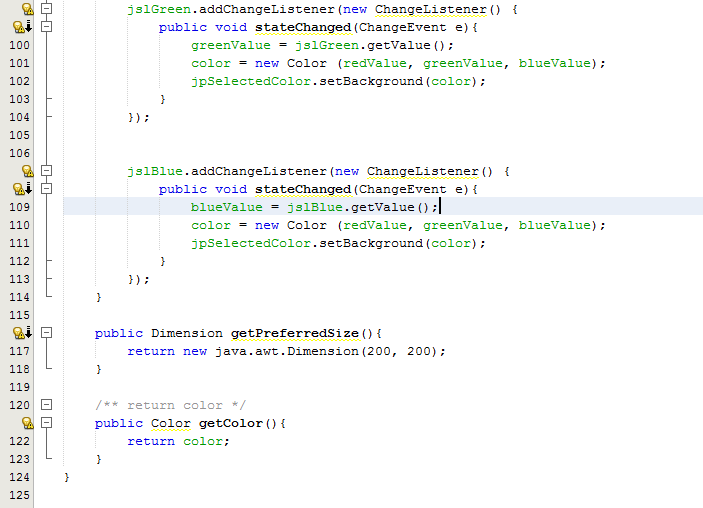
* Scenario 1: 30pts
* Scenario 2: 40pts

# Class ColorDialog









# Class Color\_Frame (using ColorDialog)

