```java
/**
 * Deborah Barndt
 * 9-6-16
 * AccountHolder.java
 * Lab 1
 * This program will similate a bank account. It will prompt users for options for
creating an initial balance, entering deposits or withdrawals, and it will
 * allow for the printing of account information including interest at various interest
rates.
 * Written by Deborah Barndt. */

package BankofIIT;

import javax.swing.JOptionPane;
import java.text.DecimalFormat;

public class AccountHolder
{
	static double balance,                                  // account balance
	              annualInterestRate = .04;       // interest rate

	//Print 2 decimal places.
	DecimalFormat mydf = new DecimalFormat("0.00");

	// Constructor sets the balance to 0.00.
	public AccountHolder()
	{
		balance = 0.00;
	}

	// Constructor sets the starting balance of the account to the value passed.
	public AccountHolder(double newbalance)
	{
		balance = newbalance;

		// Check for a positive balance
		if (balance < 0)
		{
			JOptionPane.showMessageDialog(null, "Error: You can not start off with
a negative balance.");
		}

		this.balance = balance >= 0 ? balance: 0;
	}

	// Constructor sets the balance of the account to the value as a string.
	public AccountHolder(String balstr)
	{
		balance = Double.parseDouble(balstr);
	}

	// Method to make a deposit money into the account.
	public void deposit(double amount)
	{
		balance += amount;
	}

	// Method to make a deposit money into the account as a string.
```

```
	public void deposit(String depstr)
	{
		balance += Double.parseDouble(depstr);
	}

	// Method to withdraw money from the account.
	public void withdrawal(double amount)
	{
		if (balance <= 100)
		{
			JOptionPane.showMessageDialog(null, "Insufficient Funds for
Withdrawal.");
			amount = 0;
		}

		else
		{
			// Check if the user's balance goes below $500, and add a transaction
fee.
			if (balance - amount < 500)
			{
				// Check if the user's balance is below $100.
				if (balance - amount < 100)
				{
					amount = balance - 100;
					JOptionPane.showMessageDialog(null, "Your balance cannot
go below $100.\n"
							+ "Your withdrawal is now the maximum amount
of $" + mydf.format(amount) + ".");
					balance -= amount;
					amount = 0;
				}

				balance -= 50;
				JOptionPane.showMessageDialog(null, "Your account is below
$500.\n"
						+ "A $50 transaction fee has been deducted from
your account.\n");
			}

			balance -= amount;
		}


	}

	// Method to withdraw money from the account as a string.
	public void withdrawal(String withstr)
	{
		balance -= Double.parseDouble(withstr);
	}

	// Method to calculate the monthly interest rate to the account.
	public void monthlyInterest()
	{
		balance += balance * (annualInterestRate / 12.0);
	}
```

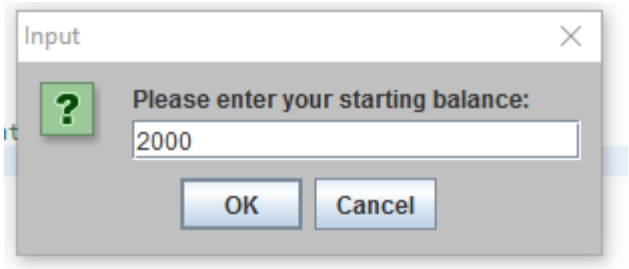
```
	// Method to modify the monthly interest rate to the account.
	public void modifyMonthlyInterest(double interestRateUpdate)
	{
		if (interestRateUpdate >= 0 && interestRateUpdate <= 1.0)
		{
			annualInterestRate = interestRateUpdate;
		}
	}

	// Setter method to set the account balance.
	public void setBalance(double newbalance)
	{
		balance = newbalance;
	}

	// Setter method that sets the account balance as a string.
	public void setBalance(String balstr)
	{
		balance = Double.parseDouble(balstr);
	}

	// Getter method to return the account balance.
	public double getBalance()
	{
		return balance;
	}

	// Getter method to return the annual interest rate.
	public static double getAnnualInterestRate()
	{
		return annualInterestRate;
	}

	// Getter method to return the accounts current balance in a string.
	@Override
	public String toString()
	{
		return String.format("$%.2f", balance);
	}

}


package BankofIIT;

import java.awt.Color;
import java.net.MalformedURLException;
import java.net.URL;
import java.text.DecimalFormat;
import javax.swing.ImageIcon;
import javax.swing.JOptionPane;
import java.text.DecimalFormat;
import javax.swing.JTextArea;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import javax.swing.UIManager;

/**
```

```
 * Deborah Barndt
 * 9-6-16
 * AccountHolderTest.java
 * Lab 1
 * This programs tests the AccountHolder class.
 * Written by Deborah Barndt. */

public class AccountHolderTest
{
	public static void main(String[] args) throws MalformedURLException
	{
		double balance,                  // balance of the account
		    annualamount;  // the annual amount added to the user's account

		int   selection = 0;      // menu selection made by the user

		//Print 2 decimal places.
		DecimalFormat mydf = new DecimalFormat("0.00");

		// Change the color of the background of the dialog box
		UIManager UI = new UIManager();
		UI.put ("OptionPane.background", Color.lightGray);
		UI.put ("Panel.background", Color.lightGray);

		// Print a time stamp for program.
		String timeStamp = new SimpleDateFormat("yyyy/MM/dd
HH:mm:ss").format(Calendar.getInstance().getTime());

		// Prompt the user for the initial balance.
		balance = Double.parseDouble(JOptionPane.showInputDialog("Please enter your
starting balance: "));

		// Create an AccountHolder object.
		AccountHolder account = new AccountHolder(balance);

		// Change icon inside dialog box.
		final ImageIcon icon = new ImageIcon(new
URL("http://iconshow.me/media/images/Business/e-commerce-icon/png/32/bank.png"));

		do
		{
			selection = Integer.parseInt(JOptionPane.showInputDialog("Welcome to
the Bank of IIT!\n"
				+ "How many I help you today?\n\n"
				+ "1: Check your current balance.\n2: Make a deposit.\n"
				+ "3: Make a withdraw.\n4: Update your interest.\n"
				+ "5: Check your 12 month annual interest.\n6: Exit\n\n"
				+ "Please make a selection: "));

			// Switch statement containing the Bank of IIT Service Menu.
			switch (selection)
			{
				case 1:
					// Display the user's current balance.

					JOptionPane.showMessageDialog(null, "Your current balance
is $" + mydf.format(account.getBalance()) + ".\n\n"
						+ "Cur dt = " + timeStamp + "\nProgrammed by
```

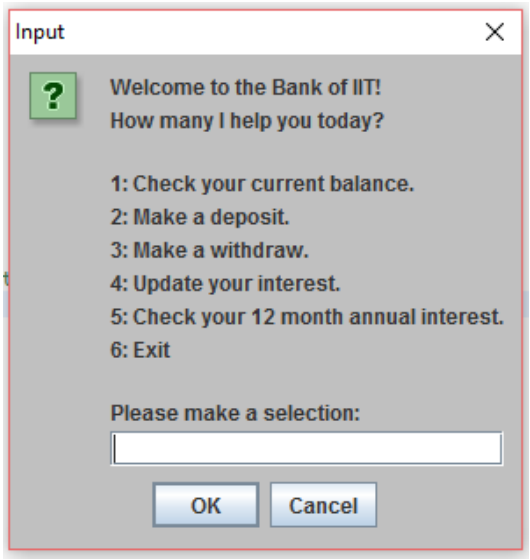
```
Deborah Barndt\n", "Account Balance",
                                                JOptionPane.INFORMATION_MESSAGE, icon);
                                    break;

                              case 2:
                                    // Prompt the user for the amount they would like to
deposit.
                                    balance =
Double.parseDouble(JOptionPane.showInputDialog("Please enter the amount you would like to
deposit: "));
                                    account.deposit(balance);
                                    JOptionPane.showMessageDialog(null, "Your current balance
is $" + mydf.format(account.getBalance()) + ".\n\n"
                                                + "Cur dt = " + timeStamp + "\nProgrammed by
Deborah Barndt\n", "Deposit",
                                                JOptionPane.INFORMATION_MESSAGE, icon);
                                    break;

                              case 3:
                                    // Prompt the user for the amount they would like to
withdrawal.
                                    balance =
Double.parseDouble(JOptionPane.showInputDialog("Please enter the amount you would like the
withdraw: "));
                                    account.withdrawal(balance);
                                    JOptionPane.showMessageDialog(null, "Your current balance
is $" + mydf.format(account.getBalance()) + ".\n\n"
                                                + "Cur dt = " + timeStamp + "\nProgrammed by
Deborah Barndt\n", "Withdraw",
                                                JOptionPane.INFORMATION_MESSAGE, icon);
                                    break;

                              case 4:
                                    // Check your current interest rate, and update the amount
of interest added to the account.

                        account.modifyMonthlyInterest(Double.parseDouble(JOptionPane.showInputDialog("Your
current interest rate is " +
                                                mydf.format(account.getAnnualInterestRate())
+ "\n"
                                                + "Input the new interest rate in decimal
format: ")));
                                    break;

                              case 5:
                                    // Display a 12 month print out showing the interest being
added to the user's account. Showing a text area containing the headers for columns,
                                    // the current balance, and the 12 month print out showing
the interest for each of the 12 months.
                                    String output = "Your current balance is $" +
mydf.format(account.getBalance()) + ".\n\n"
                                                      + "Your monthly balances for one
year at " + mydf.format(account.getAnnualInterestRate() * 100) + "% interest rate.\n\n"
                                                      + "Month\tBalance with
Interest\n";

                                    Double monthBal = account.getBalance();
```

```
                                    // Loop to calculate the 12 month print out.
                                    for (int monthNum = 1; monthNum <= 12; monthNum++)
                                    {
                                          output += monthNum + ":\t" + "$" +
mydf.format((monthBal += monthBal * (account.getAnnualInterestRate()/12))) + "\n" ;
                                    }

                                    output += "\n\n" + "Cur dt = " + timeStamp + "\nProgrammed
by Deborah Barndt\n";

                                    // Display the results.
                                    JOptionPane.showMessageDialog(null, new JTextArea(output),
"12 Month Interest Rate Print Out",
                                                JOptionPane.INFORMATION_MESSAGE, icon);

                                    break;

                              case 6:
                                    // Exit the program.
                                    System.exit(0);
                        }
                  }

                  while (selection != 6); // Ends the loop.
            }
}
```

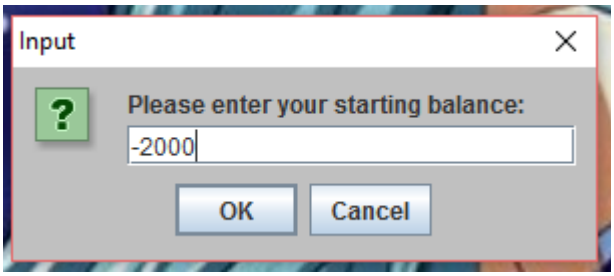
Intial starting balance:



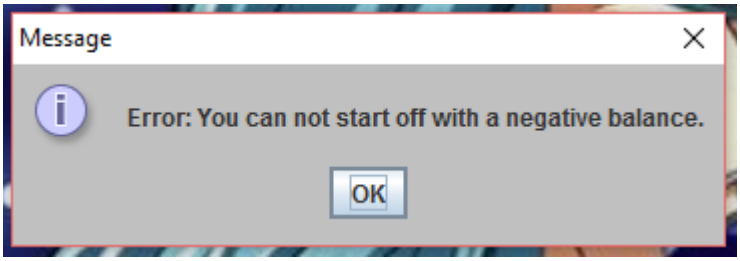
Welcome to Bank of IIT Menu:



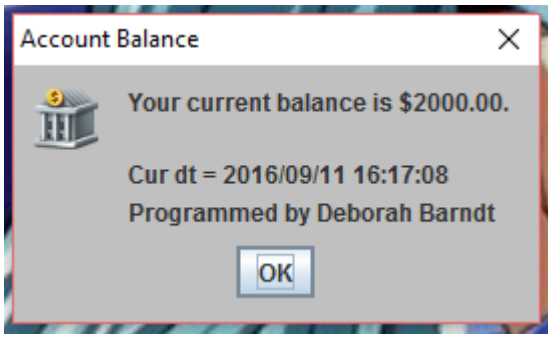
Trying to put in a negative balance:



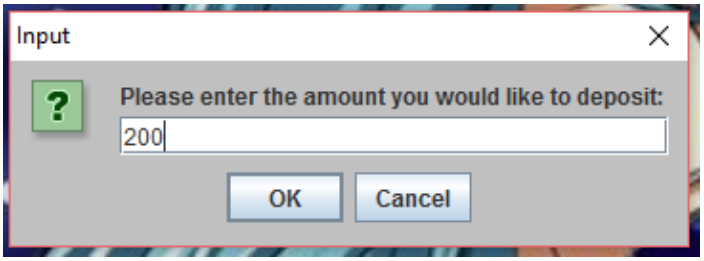
Error message for negative balance:



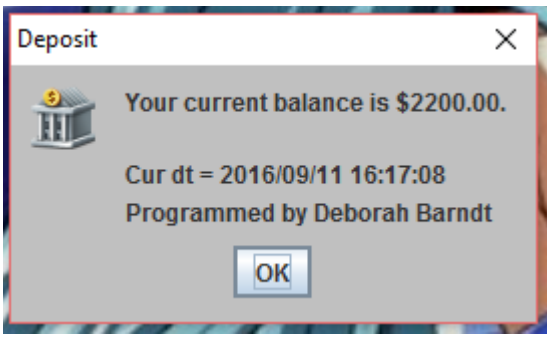
Display of current balance:



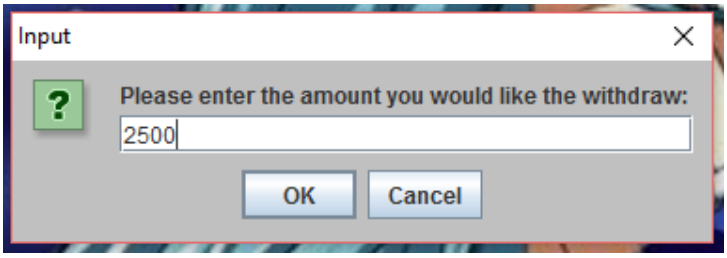
Making a deposit:



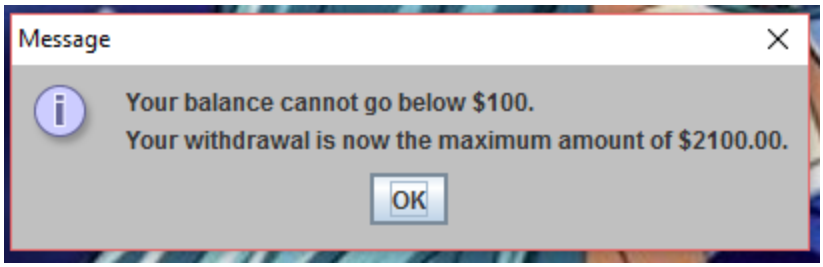
Current balance after deposit:



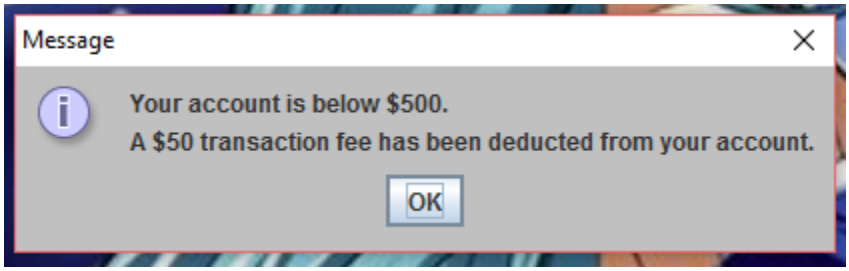
Withdrawing from account:



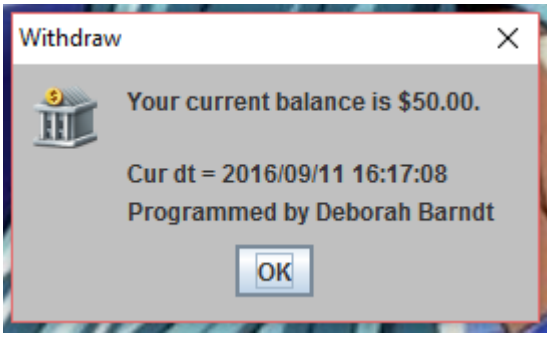
Trying to put account below 100 but does the maximum withdrawal:



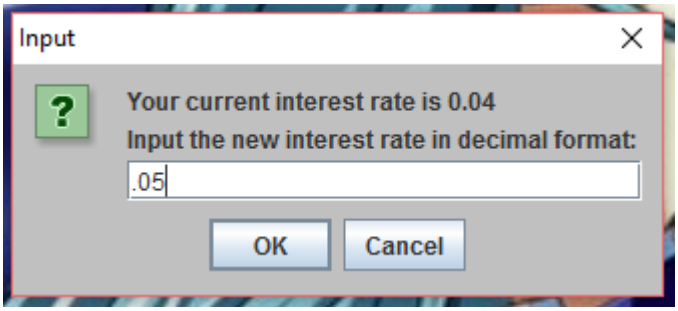
Account below 500 message:



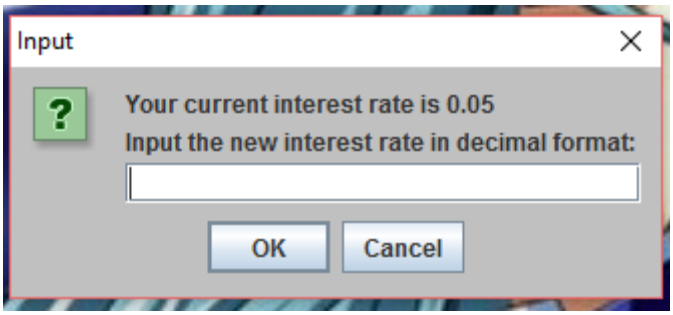
Balance after transaction fee:



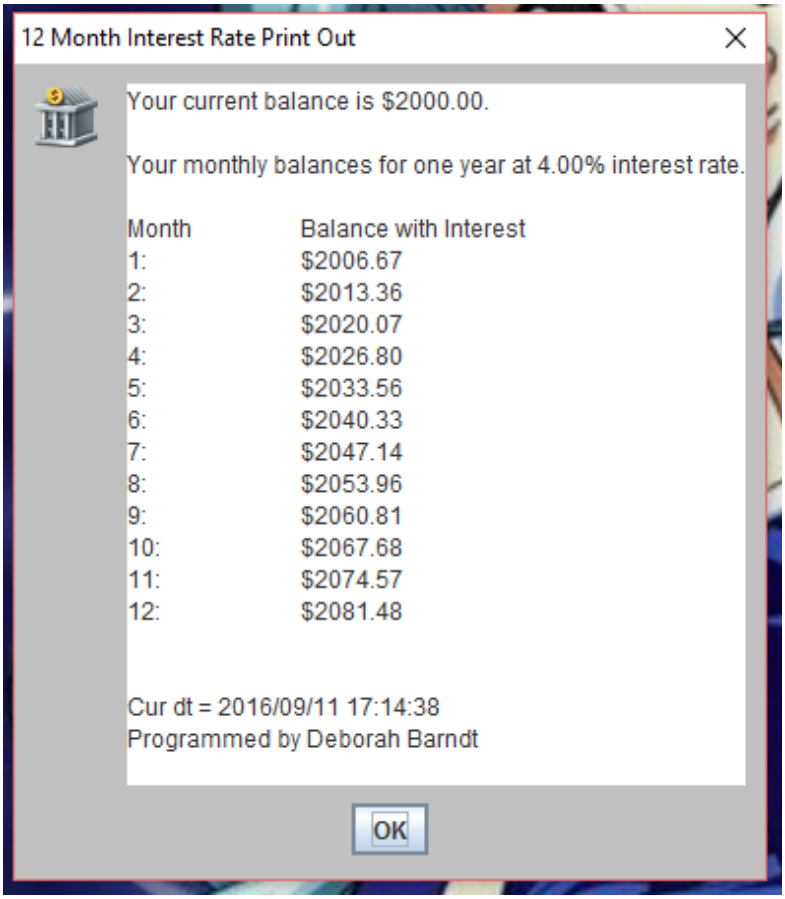
Interest rate being updated from current interest rate:



Display of updated interest rate:



Print out of the 4% interest rate print out:



Print out of the 5% interest rate print out: