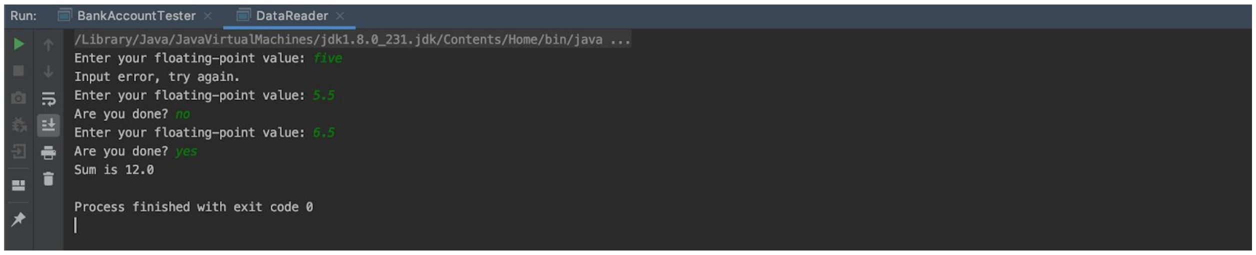
Part 1

```
/**
* Mark Fastner
* 10/28/2020
* This class reads in different floating point values and adds them together
* not-this class will catch user errors when they dont enter a folating point number
* and give the user 2 chances to input a correct value
*
*/

import java.util.Scanner;
import java.util.InputMismatchException;

public class DataReader
{
   public static void main(String[] args)
   {
       Scanner in = new Scanner(System.in);

       int chances = 0;
       boolean done = false;
       double sum = 0;
       String finish = "";
       while (!done)
       {
           double value = 0;
           try
           {  //takes in floating point value from user
               //adds it to sum
               System.out.print("Enter your floating-point value: ");
               value = in.nextDouble();
               sum = sum + value;
               chances = 0;
               System.out.print("Are you done? ");
               finish = in.next();
               if (finish.equals("Yes") || finish.equals("yes"))
                   done = true;
           }
           //catches a wrong input
           catch (InputMismatchException e)
           {
               //displays the error message
               //allows user to try again
               //adds 1 to chances
               System.out.println("Input error, try again.");
               String WrongInput = in.next();
               chances++;
               //checks if user has exceeded chances
               if (chances > 1)
               {
                   System.out.println("Wrong input again.");
                   done = true;
               }
```

```
            }
        }
        //Display the sum
        System.out.println("Sum is " + sum);
    }
}
```



```
Run:   BankAccountTester   DataReader
/Library/Java/JavaVirtualMachines/jdk1.8.0_231.jdk/Contents/Home/bin/java ...
Enter your floating-point value: five
Input error, try again.
Enter your floating-point value: 5.5
Are you done? no
Enter your floating-point value: 6.5
Are you done? yes
Sum is 12.0

Process finished with exit code 0
```

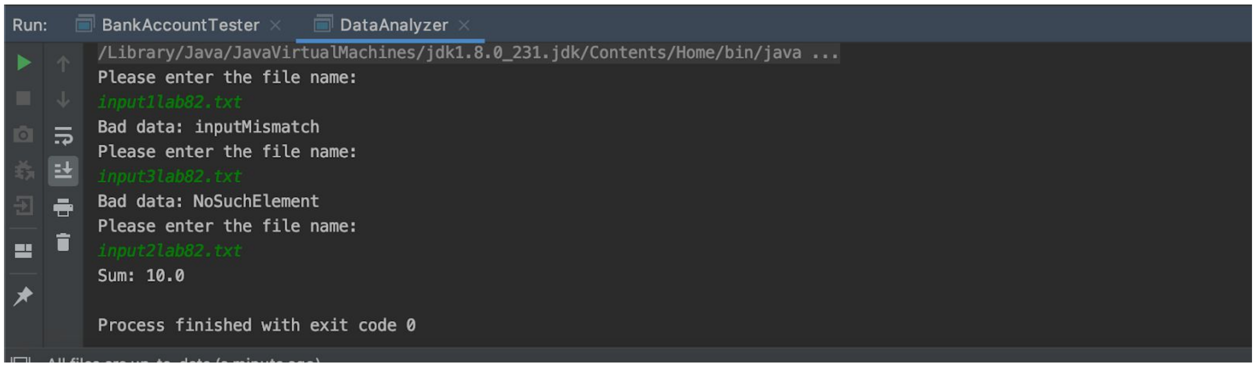
Part 2

```
import java.io.File;
import java.io.IOException;
import java.util.Scanner;
import java.util.InputMismatchException;
import java.util.NoSuchElementException;

public class DataSetReader
{
   private double[] data;

   /**
    Reads a data set.
    @param filename the name of the file holding the data
    @return the data in the file
    */
   public double[] readFile(String filename)
           throws IOException, BadDataException
   {
       try (Scanner in = new Scanner(new File(filename)))
       {
           readData(in);
       }
       catch (InputMismatchException e)
       {
           throw new BadDataException("Bad data: inputMismatch");
       }
       catch (NoSuchElementException e)
       {
           throw new BadDataException("Bad data: NoSuchElement");
       }
       return data;
   }
```

```
  /**
   Reads all data.
   @param in the scanner that scans the data
   */
  private void readData(Scanner in) throws BadDataException,
         InputMismatchException, NoSuchElementException
  {
     int numberOfValues = in.nextInt();
     data = new double[numberOfValues];
     //Use a for loop to read other inputs with readValue method
     for (int i = 0; i < data.length; i++)
     {
         readValue(in, i);
     }
     if (in.hasNext())
         throw new BadDataException("End of file expected");
  }

  /**
   Reads one data value.
   @param in the scanner that scans the data
   @param i the position of the value to read
   */
  private void readValue(Scanner in, int i)
         throws InputMismatchException, NoSuchElementException
  {   //Read a double value and store it in the array data
     data[i] = in.nextDouble();
  }
}
```

```
import java.io.FileNotFoundException;
import java.io.IOException;
import java.util.Scanner;

public class DataAnalyzer
{
  public static void main(String[] args)
  {
     Scanner in = new Scanner(System.in);
     DataSetReader reader = new DataSetReader();

     boolean done = false;
     while (!done)
     {
         try
         {
            System.out.println("Please enter the file name: ");
            String filename = in.next();

            double[] data = reader.readFile(filename);
            double sum = 0;
            for (double d : data)
            {
               sum = sum + d;
            }
            System.out.println("Sum: " + sum);
            done = true;
         }
         catch (FileNotFoundException e)
         {
            System.out.println("File not found.");
         }
         catch (BadDataException e)
         {
            System.out.println(e.getMessage());
         }
         catch (IOException e)
         {
            e.printStackTrace();
         }
      }
   }
}
```

```
/**
This class reports bad input data.
*/
public class BadDataException extends Exception
{
   public BadDataException()
   {
   }

   public BadDataException(String message)
   {
      super(message);
   }
}
```



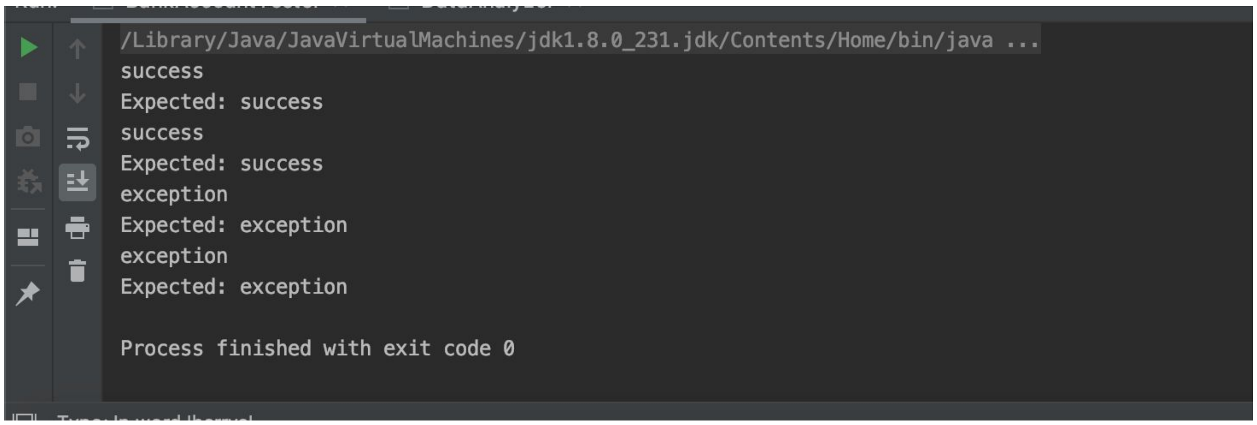
```
/Library/Java/JavaVirtualMachines/jdk1.8.0_231.jdk/Contents/Home/bin/java ...
Please enter the file name:
input1lab82.txt
Bad data: inputMismatch
Please enter the file name:
input3lab82.txt
Bad data: NoSuchElement
Please enter the file name:
input2lab82.txt
Sum: 10.0

Process finished with exit code 0
```

Part 3

```
/**
A bank account has a balance that can be changed by
deposits and withdrawals.
*/
public class BankAccount
{
   private double balance;

   /**
    Constructs a bank account with a zero balance.
    */
   public BankAccount()
   {
      balance = 0;
   }

   /**
    Constructs a bank account with a given balance.
    @param initialBalance the initial balance
    */
   public BankAccount(double initialBalance)
   {
      if (initialBalance < 0)
         throw new IllegalArgumentException(
               "Cannot create account: " + initialBalance + " is less than zero");
      balance = initialBalance;
   }

   /**
    Deposits money into the bank account.
    @param amount the amount to deposit
    */
   public void deposit(double amount)
   {
      if (amount < 0)
         throw new IllegalArgumentException(
               "Deposit of " + amount + " is less than zero");
      double newBalance = balance + amount;
      balance = newBalance;
   }
   /**
    Withdraws money from the bank account.
    @param amount the amount to withdraw
    */
   public void withdraw(double amount)
   {
      if (amount < 0 || amount > balance)
         throw new IllegalArgumentException(
               "Withdrawal of " + amount + " is less than zero or Withdrawal of "
                     + amount + " exceeds balance of " + balance);

      double newBalance = balance - amount;
      balance = newBalance;
   }

   /**
    Gets the current balance of the bank account.
    @return the current balance
    */
   public double getBalance()
   {
      return balance;
   }
}
```

```
/**
A class to test the BankAccount class.
*/
public class BankAccountTester
{
   public static void main(String[] args)
   {
      BankAccount harrysChecking = new BankAccount();

      try
      {
         harrysChecking.deposit(300);
         System.out.println("success");
      }
      catch (IllegalArgumentException e)
      {
         System.out.println("exception");
      }
      System.out.println("Expected: success");

      try
      {
```

```java
            harrysChecking.withdraw(100);
            System.out.println("success");
        }
        catch (IllegalArgumentException e)
        {
            System.out.println("exception");
        }
        System.out.println("Expected: success");

        try
        {
            harrysChecking.deposit(-100);
            System.out.println("success");
        }
        catch (IllegalArgumentException e)
        {
            System.out.println("exception");
        }
        System.out.println("Expected: exception");

        try
        {
            harrysChecking.withdraw(300);
            System.out.println("success");
        }
        catch (IllegalArgumentException e)
        {
            System.out.println("exception");
        }
        System.out.println("Expected: exception");
    }
}
```



```
/Library/Java/JavaVirtualMachines/jdk1.8.0_231.jdk/Contents/Home/bin/java ...
success
Expected: success
success
Expected: success
exception
Expected: exception
exception
Expected: exception

Process finished with exit code 0
```

Part 4

```
/**
A bank account has a balance that can be changed by
deposits and withdrawals.
*/
public class BankAccount2
{
   private double balance;

   /**
    Constructs a bank account with a zero balance
    */
   public BankAccount2()
   {
      balance = 0;
   }

   /**
    Constructs a bank account with a given balance
    @param initialBalance the initial balance
     * @throws NegativeBalanceException
    */
   public BankAccount2(double initialBalance) throws NegativeBalanceException
   {
      if (initialBalance < 0)
         throw new NegativeBalanceException(
               "Cannot create account: " + initialBalance + " is less than zero");

      balance = initialBalance;
   }

   /**
    Deposits money into the bank account.
    @param amount the amount to deposit
     * @throws NegativeAmountException
    */
   public void deposit(double amount) throws NegativeAmountException
   {
      if (amount < 0)
         throw new NegativeAmountException(
               "Deposit of " + amount + " is less than zero");
      double newBalance = balance + amount;
      balance = newBalance;
   }

   /**
    Withdraws money from the bank account.
    @param amount the amount to withdraw
     * @throws NegativeAmountException
    */
   public void withdraw(double amount) throws NegativeAmountException, InsufficientFundsException
   {
      if (amount < 0)
         throw new NegativeAmountException(
                  "Withdrawal of " + amount + " is less than zero");

      if (amount > balance)
          throw new InsufficientFundsException(
                  "Withdrawal of " + amount + " exceeds balance of " + balance);

      double newBalance = balance - amount;
      balance = newBalance;
   }

   /**
    Gets the current balance of the bank account.
    @return the current balance
    */
   public double getBalance()
   {
      return balance;
   }
}
```

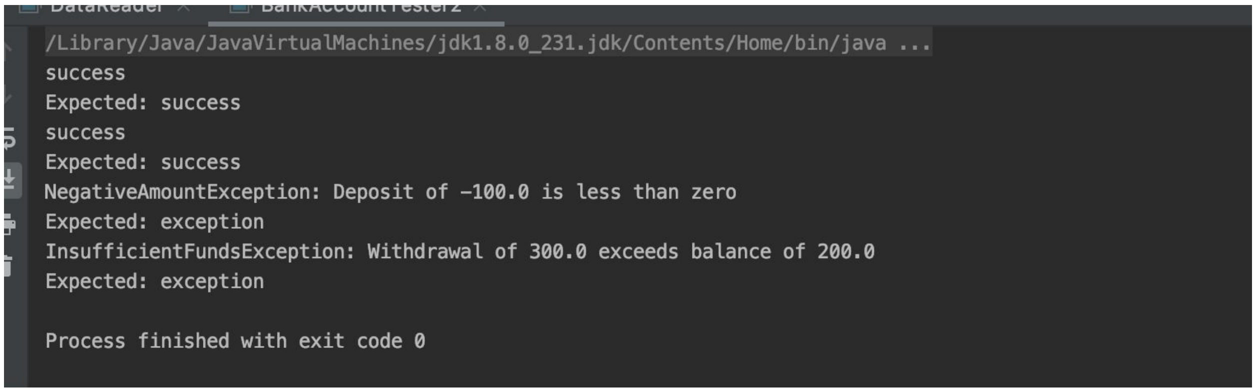
```
/**
A class to test the BankAccount class.
*/
public class BankAccountTester2
{
   public static void main(String[] args)
   {
      BankAccount2 harrysChecking = new BankAccount2();
      try
      {
          harrysChecking.deposit(300);
          System.out.println("success");
      }
      catch (NegativeAmountException e)
      {
          System.out.println(e);
      }
      System.out.println("Expected: success");
      try
      {
          harrysChecking.withdraw(100);
          System.out.println("success");
      }
      catch ( InsufficientFundsException | NegativeAmountException e)
      {
          System.out.println(e);
      }
      System.out.println("Expected: success");
      try
      {
         harrysChecking.deposit(-100);
         System.out.println("success");
      }
      catch (NegativeAmountException  e)
      {
         System.out.println(e);
      }
      System.out.println("Expected: exception");

      try
      {
         harrysChecking.withdraw(300);
         System.out.println("success");
      }
      catch ( InsufficientFundsException | NegativeAmountException e)
      {
         System.out.println(e);
      }
      System.out.println("Expected: exception");
   }
}
```

```
public class InsufficientFundsException extends Exception
{
   public InsufficientFundsException()
   {
   }

   public InsufficientFundsException(String message)
   {
      super(message);
   }
}
```

```
public class NegativeAmountException extends Exception
{
   public NegativeAmountException()
   {
   }
```

```java
    public NegativeAmountException(String message)
    {
        super(message);
    }
}
```

```java
public class NegativeBalanceException extends Exception
{
    public NegativeBalanceException()
    {
    }

    public NegativeBalanceException(String message)
    {
        super(message);
    }
}
```



```
/Library/Java/JavaVirtualMachines/jdk1.8.0_231.jdk/Contents/Home/bin/java ...
success
Expected: success
success
Expected: success
NegativeAmountException: Deposit of -100.0 is less than zero
Expected: exception
InsufficientFundsException: Withdrawal of 300.0 exceeds balance of 200.0
Expected: exception

Process finished with exit code 0
```