

IST 256  
Lab  
Week 4 – February 10, 2010

### 1. Compute Interest – extending the compute interest program

In the lab portion of class on Monday, we created a project called ComputeInterest with a GUI application called ComputeInterestForm. In this first exercise, we are going to extend this application so that the user can specify to save an additional amount each year.

Start NetBeans and open the ComputeInterest project that you created on Monday. To open the project, either click on the triangle by the project name in the project pane, or go to the File -> OpenProject menu and select the project name. In either case, next open the triangle by the Source Packages, then default packages and then double click on ComputeInterestForm.java. Now your code should go into the main pane and you can select between the Source and Design tabs.

In this application, the user typed in numbers for the following:

Initial bank balance

Number of years to compute a new balance

There was a button to Compute Interest and a textfield to display the results.

When the user clicked on the Compute Interest button, a new bank balance was computed by computing the interest each year at 5% and adding it into the bank balance for that year.

Here is the program that you put in the actionPerformed method (making sure that the opening and closing { } matched). Note that your TextFields or Labels may be numbered differently than the one shown here.

```
double balance, yearlyinterest = 0.0;
int numyears = 0;

// get savings amount and number of years from user
balance = Double.parseDouble(jTextField1.getText());
numyears = Integer.parseInt(jTextField2.getText());

// compute 5% interest each year and add to balance
for (int i = 1; i <= numyears; i++) {
    yearlyinterest = balance * .05;
    balance = balance + yearlyinterest;
}

// display the new balance to the user
jTextField3.setText(String.valueOf(balance));
```

For lab today, you are to extend this Compute Interest application. In the extended version, the user can also type in an amount that he or she proposes to add to the bank balance as an investment each year.

Add to the Form an instruction label and a TextField for the user to type in their annual investment (and remember the number of the TextField).

Add to the code in the actionPerformed method for the button:

- Add a declaration of a variable of type double to hold the investment amount.
- Add the statement that gets the investment value from the TextField and assigns it to the variable.
- Change the code that computes the new balance every year, assuming that you called the investment amount, investment, to:
  - $\text{balance} = \text{balance} + \text{yearlyinterest} + \text{investment}$

Test your program and write one example here:

Initial bank balance typed by user	
Number of years	
Annual investment	
New bank balance computed	

## 2. Understanding While Loops – showing the variables of a program

For this problem, you are not to put any part of the program into NetBeans. You are to write your answers on this lab sheet.

A. Consider the following program:

```
int sum = 0;
int count = 1;

//sum the integers starting from 1
//    until the sum is greater than or equal 12
while (sum < 12)
{
    sum = sum + count;
    count++;
}
System.out.println("Numbers counted = " + count);
```

As this program is executed, it will keep the variables in memory, and the values of the variables will change during the execution of the program.

1. In the table below, write the values of the variables count and sum after they are initialized (in the first assignment statement) under the column labeled “init”.
2. Then for each loop iteration, write the values of the variables count and sum at the end of that loop iteration (from the assignment statements during the loop. These are labeled Val1 on up in the table. At each iteration, **decide whether the loop continues** and only put in as many values as necessary:

Variable	init	Val1	Val2	Val3	Val4	Val5	Val6	Val7
counter								
sum								

What number will be shown to the user in Label1.Text at the end of the program? (write it here):

#### 4. Compute Savings

For this application, you are going to create a similar application to the one that you did in the last example, but using a while loop.

Now open a new project with NetBeans named something like **ComputeSavings**. This new project should be in the same lab projects folder as usual.

**For this example**, you are to write a similar application to the ComputeInterest project, except that this time the user wants to achieve a savings goal and needs to compute how many years it will take to achieve their goal amount.

Write this application, allowing the user to type in:

Initial bank balance

Annual investment

**Goal savings amount**

There should be a button named something like Compute Number of Years and a TextField or Label to show the results.

You may want to refer back to the code for the ComputeInterest program.

- Declare variables for balance, yearly interest, investment, goal and numberyears.
- After the declarations of variables, set the numberyears variable to start at 0.
- Write a while loop.
  - The condition for the while loop should be to continue the loop as long as the balance is less than the savings goal.
  - The body of the loop will include the same statements as for ComputeInterest, computing the yearlyinterest and the new balance during each iteration of the loop.
  - But also add a statement to the body of the while loop to explicitly add one to the numberyears variable during the loop.
- After the while loop, display the number of years to the user.

Test your program and write one example here:

Initial bank balance typed by user	
Annual investment	
Savings goal	
Number of years to achieve goal	

**Be prepared to answer questions about your program and your results.**