**Name:** Hamna Zahid

**Reg. No.** : 33787

**Summer 2017**

**Advanced Programming**

**Assignment # 1 : Simple Interpreter**

**Date:** 23-1-2017 (Using one late day)

**Introduction:**

In this assignment I have developed an interpreter in java which interprets the code given i.e. executes the given code, according to the rules mentioned in the assignment. In case of invalid code, it prints the error along with the line number on which it occurs.

I have also written unit tests for all the rules mentioned. There are also two unit tests which read code from text files (the examples given in the assignment) and test the program. All the tests have been passed.

The results of processor and memory profiling have been included in the document.

**Novelty:**

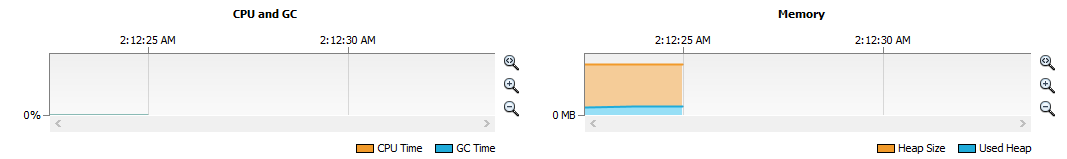
I have learnt how to do CPU, memory, garbage collection and threads and classes profiling in net beans.

Plus I also learnt how to make unit tests while making unit tests for the interpreter rules. Especially the tests don’t run if your code run system.exit() for terminating the program (program is terminating in case of invalid code after printing the error). So I had to use a separate library ‘ExpectedSystemExit’ for that.

I have also learnt how to use maps for storing functions. I have used identifiers in code as keys and functions as values of maps. For e-g: ‘+’ is a key in the map and the function ‘addition()’ is the corresponding value.

I have also used maps to store the variables in the code. I have made two classes Int and Bool to represent the datatypes of the variables. Int consists of a signed 64 bit integer (long in java) while Bool can either be tt or ff. These both inherit from class DataType which has a variable for storing the type names.

**Memory and processor profiling:**



**Assumptions:** (Not contradicting with the assignment)

* All the identifiers, operands and operators should have space between them.
* tt represents true, ff represents false
* Names start with only alphabets and may contain the special characters mentioned in the assignment.
* ‘- a’ inverts the value of ‘a’ but does not assign the inverted value to ‘a’ in case of not using assignment operator.
* Loosely typed language which means that we don’t specify the datatype while declaring a variable, the type gets assigned on the basis of the assignment operation.

**How to run:**

I have my net beans project solution folder inside zip folder which can be executed by extracting it and running with the help of compiler. The code text files have been included in the same folder as that containing the src folder.

**Screen shots of sample runs:**

**Code example 1:**

Code1.txt

Let x

Let y = 0 ;

x = y ;

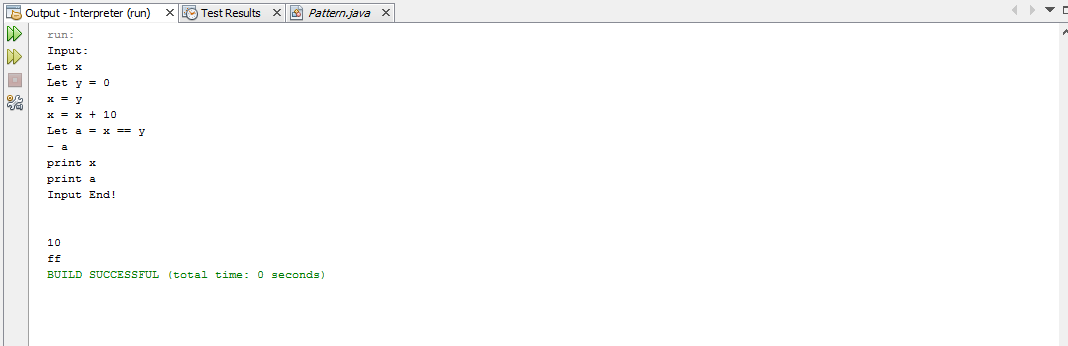
x = x + 10 ;

Let a = x == y;

- a

print x

print a



**Code example 2:**

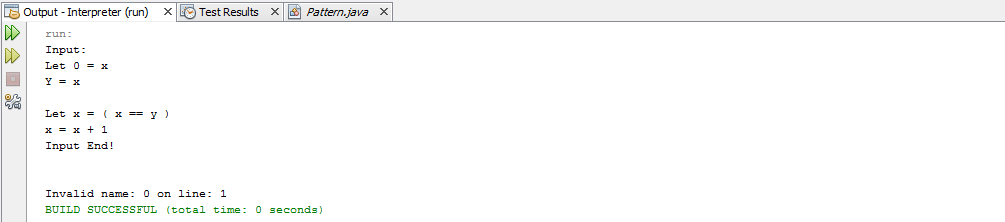
Code2.txt:

Let 0 = x;

Y = x;

Let x = ( x == y );

x = x + 1;



**Code example 3:**

Code3.txt

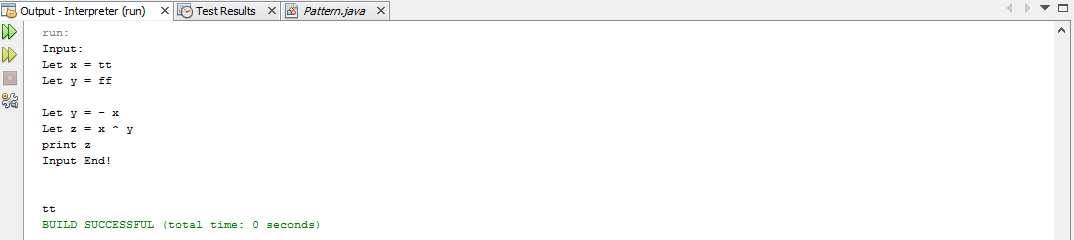
Let x = tt;

Let y = ff;

Let y = - x;

Let z = x ^ y;

print z;



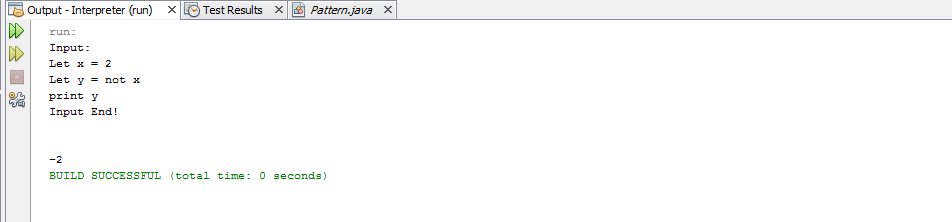
**Code example 4:**

Code 4.txt:

Let x = 2;

Let y = not x;

print y;



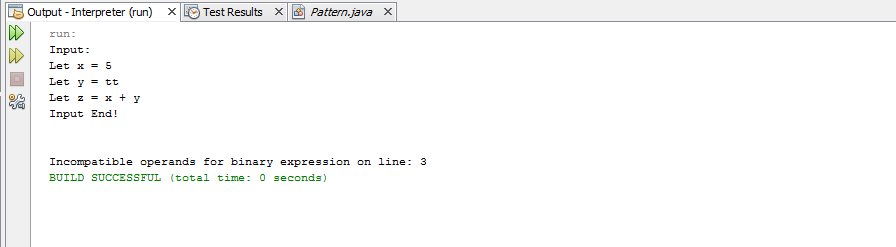
**Code example 5:**

Code5.txt:

Let x = 5

Let y = tt;

Let z = x + y;



**Result from running all 11 unit tests:**

