Midterm Instructions

MY NAME: ..............................................
MY NYU ID: ............................................
MY EMAIL: ..............................................

Please read carefully:

0. This exam is worth 135 Points
1. Do all questions.
2. Write answers in the provided spaces.
   If necessary, programs can appear on a reverse white page.
3. BIG HINT: sketch out your answer in the margin or the
   reverse white page BEFORE filling in the answers.
4. This is a closed book exam (no calculators or computers).
   You may refer to one “cheat sheet” (8” by 11”, 2-sided).
5. Always provide a brief justifications (even for T/F questions)
6. Please allocate your time wisely.
7. Good Luck!

MY SCORES: (for official use only)

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>TOTAL</th>
</tr>
</thead>
</table>
QUESTION 1.

SHORT QUESTIONS (5 Points each part = 50 Points).

(a) (5 Points for each error)
This Java program has two errors.

```java
public class FixMe {
    int nn;
    nn = 111;
    public static void main (String[] args) {
        System.out.printf("nn = %d\n", nn);
    }
}
```

Describe the two errors. For each error, state whether it is a run-time or compile-time error.

**ANSWER:**

(b) Please remove both errors in FixMe

**ANSWER:**

(c) Using the geometric shape examples from Chapter 1, consider:

```java
public void changeParameter (Circle c) {
    c.setRadius(5);
}

public static void main(String[] args) {
    Circle a = new Circle();
    a.setRadius(10);
    Circle b = a;
    changeParameter(b);
}
```

What are the radii of circle a and circle b at the end?

**ANSWER:**

(d) Suppose we have doubly-linked lists whose nodes are instances of
class Node {
    String val;
    Node next, prev;
}

Write the method
    void remove(Node u) which will remove the node that precedes u. If u or u.prev is null, then this code does nothing.

ANSWER:

(e) Suppose the running time for an algorithm satisfies this recurrence

\[ T(n) = \begin{cases} 
4 & \text{for } n \leq 2 \\
10 + 3T(n - 1) & \text{else}.
\end{cases} \]

Prove that \( T(n) = 3^n - 5 \) is a solution.

ANSWER:

(f) Here is a partial listing of my Makefile:

```
jcc:
    ----> test -d bin || mkdir bin
    ----> -javac -d bin src/*//*.java

hw1:
    ----> make p=hw1 hw1-helper

hw1-helper:
    ----> java src/hw1/myProg $(args)
```

As usual ----> just indicates a tab. Describe all the actions of the jcc target.

ANSWER:
(g) In the Makefile above, what is the point of target hw1 calling another target hw1-helper? Why not call hw1-helper directly? HINT: our two goals are to set up default arguments that depend upon the target (hw1, hw2, etc) and the ability to override these values from the terminal.

**ANSWER:**

(h) We want a pattern that is matched by names that are written following certain common conventions (assuming only two-part names). More precisely: write any pattern that is matched by these three strings

“Firstname Lastname”, “First Last”, “Chee Yap”

but not by any of these five strings

“First Middle Last”, “First LAST”, “First Last “, “CheeYap”, “first last”

Note that we allow spaces before the name, but not after the name.

**ANSWER:**

(i) Write a Java pattern expression to convert the string

ss = "hw1.java hw2.javac zoombini.java fibjava java.class fib.java"

into

"hw1.class hw2.javac zoombini.class fibjava java.class fib.class"

**ANSWER:**

QUESTION 2. (2+10+4+12+20 = 47 Points)

The function $G(n)$ is recursively defined by the formula

$$G(n) = \begin{cases} 
1 & \text{for } n \leq 2 \\
G(n - 1) + G(n - 2) + G(n - 3) & \text{else.}
\end{cases}$$

Points are deducted for Java syntax errors in the programming parts.

(a) What is $G(6)$?

**ANSWER:**

(b) Implement a non-exponential time method to compute $G(n)$ based on recursion. Please use the following header for your method:

```java
BigInteger G(int n);
```

HINT: call a helper method to do the recursive work.
(c) What is the big-Oh running time of your method?

**ANSWER:**

(d) The value \( G(n) \) will be a very large integer even for modest values of \( n \). We would like to only see the last few digits of the number (this is enough to visually check that our computation is correct). To accomplish this, write the following method

```java
String truncate(BigInteger N)
```

that returns a String of length at most 5, representing the last 5 digits of the integer \( N \). In case \( N \) has more than 5 digits, your output string must be preceded by 3 asterisks, as in "***12345". HINT: \( N \).remainder(M) will return the remainder of \( N \) divided by the BigInteger \( M \).

**ANSWER:**

(e) Write a class called GG containing the above methods. Include a main method that takes two optional command line arguments: \texttt{int nn, int mm}. Their default values are \( nn=1000, \ mm=4 \). The goal is to compute a table of \( mm \) consecutive values of the function \( G(n) \), starting from \( n \) equal to \( nn \). Try to format the output table nicely. This problem requires file manipulation because you are to write your table into a file.
called GTABLE.
HINT: Do not forget any necessary Java imports.
PLEASE THE BACK OF THIS PAGE TO WRITE YOUR CODE.

QUESTION 3. (3+5+8+24=40 Points) Here is the stack interface from the textbook:

```java
public interface StackInterface<T> {
    void pop() throws StackUnderflowException;
    T top() throws StackUnderflowException;
    boolean isEmpty();
}
```

(a) Explain why the push operation is missing in this interface for a stack?

**ANSWER:**

(b) Write an extension of `StackInterface` with a push method that is suitable for an linked list implementation.

**ANSWER:**

(c) Write an extension of `StackInterface` with a push method that is suitable for an array implementation. Besides adding a push method, think of an additional method that would be useful to have in your extension.

**ANSWER:**

(d) Write a class called `MyLinkedStack` that implements your interface in part(b) with `T=String`. You must include a main method (no need for any command line arguments). The method is to do some pushes and pops of random strings, in order to exercise all your methods. How to get the random strings? Generate random integers and convert them into strings. Again, do not forget any needed imports.

PLEASE USE THE BACK OF THIS PAGE TO WRITE YOUR CODE.