German University in Cairo
Faculty of Media Engineering and Technology
Prof. Dr. Slim Abdennadher

April 9, 2011

Introduction to Computer Programming
Spring term 2011
Midterm Exam

Bar Code

Instructions: Read carefully before proceeding.

1) Duration of the exam: 2 hours (120 minutes).
2) (Non-programmable) Calculators are allowed.
3) No books or other aids are permitted for this test.
4) This exam booklet contains 10 pages (5 exercises), including this one. Three extra sheets of scratch paper are attached and have to be kept attached. **Note that if one or more pages are missing, you will lose their points. Thus, you must check that your exam booklet is complete**.
5) Write your solutions in the space provided. If you need more space, write on the back of the sheet containing the problem or on the three extra sheets and make an arrow indicating that. **Scratch sheets will not be graded unless an arrow on the problem page indicates that the solution extends to the scratch sheets**.
6) When you are told that time is up, stop working on the test.

Good Luck!

Don't write anything below `;-)`

Exercise	1	2	3	4	5	$\sum$
Possible Marks	6	12	10	10	12	50
Final Marks						

Exercise 1 (2+4=6 Marks)

Tracing and Nested Loops

Given the following code:

```
int n = 3;
for (int i = 0; i < n; i++)
    for (int j = 0; j < n; j++)
         System.out.println(i + " " + j);
```

a) What will be displayed on the screen when the above code is executed? Briefly explain your answer.

Solution:

```
0 0
0 1
0 2
1 0
1 1
1 2
2 0
2 1
2 2
```

b) Fill in the blanks to get an equivalent code:

```
int n = 3;
for (int i = .....; i < .......; ....)
     System.out.println(...... + " " + .....);
```

Solution:

```
int n = 3;
for (int i = 0; i < n*n; ++i)
  System.out.println(i / 3 + " " + i % 3);
```

Exercise 2 (5+7=12 Marks)

Switch + Loop

You are driving a car, you start by heading north, you can:

- `'F'` move forward
- `'U'` make a U-turn
- `'L'` turn left
- `'R'` turn right

a) Given a character `'F'`, `'U'`,`'L'` or `'R'`, write a Java program that determine your direction either north, south, east or west. For example, since you are heading north, if you are making a U-turn, then your direction will be south.

You are only allowed to use switch statements.

Solution:

```
class Ex2a {
  public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    char step = sc.next().charAt(0);
    switch (step) {
      case 'F': System.out.println("North"); break;
      case 'U': System.out.println("South"); break;
      case 'L': System.out.println("West"); break;
      case 'R': System.out.println("East");
    }
  }
}
```

b) Given a string of actions, write a Java program that will determine your final direction.

Solution:

```
class Ex2bSol1 {
  public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    int dir = 0; // 0 - North, 1 - East, 2 - South, 3 - West
    String actions = sc.next();
    for (int i = 0; i < actions.length(); ++i) {
      switch (actions.charAt(i)) {
        // 'F' does not change direction
        case 'U': dir = (dir + 2) % 4; break;
        case 'L': dir = (dir + 3) % 4; break; // Tricky!! dir = (dir - 1) % 4 will not   rk
        case 'R': dir = (dir + 1) % 4;
      }
    }
    switch (dir) {
      case 0: System.out.println("North"); break;
      case 1: System.out.println("East"); break;
      case 2: System.out.println("South"); break;
      case 3: System.out.println("West");
    }
  }
}
```

A second solution:

```
class Ex2bSol2 {
  public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    char dir = 'N';
    String actions = sc.next();
    for (int i = 0; i < actions.length(); ++i) {
      switch (actions.charAt(i)) {
        // 'F' does not change direction
        case 'U':
          if (dir == 'N') dir = 'S';
          else if (dir == 'S') dir = 'N';
          else if (dir == 'E') dir = 'W';
          else dir = 'E';
          break;
        case 'L':
          if (dir == 'N') dir = 'W';
          else if (dir == 'S') dir = 'E';
          else if (dir == 'E') dir = 'N';
          else dir = 'S';
          break;
        case 'R':
          if (dir == 'N') dir = 'E';
          else if (dir == 'S') dir = 'W';
          else if (dir == 'E') dir = 'S';
          else dir = 'N';
      }
    }
    System.out.println(dir);
  }
}
```

Exercise 3 (8+2=10 Marks)

Method + Simple Loop

Given an integer n, keep repeating the following and stop until `n=1`:

```
if n is even:
  n = n / 2
else:
  n = 3n + 1
```

a) Write a Java method that takes an integer `n` as input and prints the sequence according to the algorithm described above.

b) Write a `main` method to test the method above.

Solution:

```
import java.util.*;

class Ex3 {
  public static void printSequence(int n) {
    while (n != 1) {
      System.out.println(n);
      if (n % 2 == 0) n = n /2;
      else n = 3 * n + 1;
    }
    System.out.println(1); // Optional, it was not clear from the question whether the 1 is inclu
  }

  public static void main(String args[]) {
    Scanner sc = new Scanner(System.in);
    printSequence(sc.nextInt());
  }
}
```

Predefined Java Methods

For the next two exercises (4 and 5), you can use the following methods:

a) The method `String substring(int beginIndex)` returns a new string that is a substring of this string. The substring begins with the character at the specified index and extends to the end of this string.

For example:

```
"unhappy".substring(2) returns "happy"
"Harbison".substring(3) returns "bison"
"emptiness".substring(9) returns "" (an empty string)
```

b) The method `String substring(int start, int end)` returns a new string that is a substring of this string. The first integer argument `start` specifies the index of the first character. The second integer argument `end` is the index of the last character - 1.

For example

```
"hamburger".substring(4, 8) returns "urge"
"smiles".substring(1, 5) returns "mile"
"hello".substring( 1 ,3 ) returns "el"
```

Exercise 4 (10 Marks)

Simple Loop

A path, the general form of a filename or of a directory name, specifies a unique location in a file system. A path points to a file system location by following the directory tree hierarchy expressed in a string of characters in which path components, separated by a delimiting character, represent each directory. The delimiting character is most commonly the slash "/" [Wikipedia].

An absolute path is a path that points to the same location on one file system regardless of the working directory or combined paths. It is usually written in reference to a root directory [Wikipedia].

Given an absolute path to a file, write a Java program that prints the file name and directory.

Example: If the absolute path is `"/home/foo/bar"`

then the filename is `"bar"` and the directory is `"/home/foo/"`

Solution:

```
import java.util.*;
class Ex4 {
  public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    String path = sc.next();
    int i;
    for (i = path.length() - 1; path.charAt(i) != '/'; --i); // or instead of ";", "{}"
    System.out.println(path.substring(0, i + 1)); // print filename
    System.out.println(path.substring(i + 1)); // print directory
  }
}
```

Exercise 5 (12 Marks)

Nested Loops

Web log analysis procedures work usually on the reverse of domains.

Write a Java program that takes a domain as a string and returns a string containing the reversed domain.

For example: if the domain is "www.google.com.eg" then the program should display "eg.com.google.www".

Assuming that the domain is never empty, use a do while loop in your program.

Solution:

```
    public static String revDomain(String dom)
    {
     int i = dom.length();
         int pointer = dom.length(); //points to the beginning of the word
     String rev = "";

     while(i>0)
     {
     do
     {
     pointer--;
     }while(pointer>=0 && dom.charAt(pointer)!='.');

     rev+=dom.substring(pointer+1,i);
                if(pointer>0)   //if it is not the last word
                    rev+='.';
                i = pointer;
     }
     return rev;
    }
```

A second solution:

```
import java.util.*;
class Ex5 {
  public static void main(String args[]) {
    Scanner sc = new Scanner(System.in);
    String domain = sc.next();

    String result = "";
    int i = -1;
    do {
      // Every time the loop begins, i must be the index of a "."
      int firstChar = i + 1;
      for (i++; i < domain.length() && domain.charAt(i) != '.'; ++i);
      result = domain.substring(firstChar, i) + "." + result;
    } while (i < domain.length());
    System.out.println(result.substring(0, result.length() - 1));
  }
}
```

Extra Sheet

Extra Sheet

Extra Sheet