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

April 17, 2013

Introduction to Computer Programming
Spring term 2013
Final 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 12 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	12	14	8	12	14	60
Final Marks						

Exercise 1 (1+1+1+1+3+5=12 Marks)

a) What reserved word can be used to end an execution of a loop?

Solution:

break

b) What is the value of `var` after the following code is executed?

```
int var = 2;
switch(var) {
   case 1: var += 1;
   case 2: var += 3;
   case 3: var += 5;
}
```

Solution:

10

c) If the program compiles correctly, how many times will `Hello` be printed? Justify your answer. Highlight syntax errors if any.

```
public static void main(String[] args) {
while(true);
  {
    System.out.println("Hello");
  }
}
```

Solution:

Syntax error. Unreachable code after the condition of the while.

d) If the program compiles correctly, how many times (if any) will `Hello` be printed? Justify your answer. Highlight syntax errors if any.

```
public static void main(String[] args) {
do
  {
    System.out.println("Hello");
  }
while(true)
}
```

Solution:

Syntax error. Missing ; after the while condition.

e) The following `while` loop

```
public static void main(String[] args) {
int sum = 0;
int i = 3;
while (i < 10) {
   i++;
   if(i % 3 == 0)
     sum += i;
}
}
```

is converted into the `for` loop below. What is wrong? Correct it.

```
public static void main(String[] args) {
int sum = 0;
for (int i = 3; i < 10; i++) {
    if (i % 3 == 0)
       sum += i;
}
}
```

Solution:

Incrementing the i before the if condition in the while loop. It has to be after the if condition to have the same sum.

f) For each of the Java expressions below, write down the type of the expression and its value. If the expression causes a syntax or run-time error, justify your answer

1. `1 + 2 + "3" + 4 + 5`

   Solution:
   Type: String , Value: 3345

2. `(double)(1 / 2 + 1.0)`

   Solution:
   Type: double ,Value: 1.0

3. `false && (!(!!true || (true || !true)))`

   Solution:
   Type :boolean, Value: false

4. `7 = 11`

   Solution:
   Syntax error. The left-hand side of an assignment must be a variable.

5. `true != false`

   Solution:
   Type: boolean, Value: true

Exercise 2 (5+9=14 Marks)

a) A sequence of six tests, all scored out of 100, are to be given different weightings in determining a final mark. Write a Java program that computes the appropriate weighted score for one test. The fragment should first read values of `testNumber` and `score`. Using a switch statement, it should then compute and print the appropriate value of `weightedScore` using the weightings given in the following table.

Test Number	Weight
1	10%
2	20%
3	20%
4	15%
5	15%
6	20%

For example, input of 3 and 27 should produce the following output:

```
A score of 27 on test 3 gives a weighted score of 5.4.
```

You are neither allowed to replicate statements nor to use more than 2 print statements.

```
public class Score {

public static void main(String[] args) {
```

Solution:

```
        Scanner sc = new Scanner(System.in);
        System.out.println("please enter the test number and the score");
        int testNumber = sc.nextInt();
        int score = sc.nextInt();
        double weight = 0;
        switch (testNumber) {
        case 1:
            weight = 0.1 * score;
            break;
        case 2:
        case 3:
        case 6:
            weight = 0.2 * score;
            break;
        case 4:
        case 5:
            weight = 0.15 * score;
            break;
        default:
            weight = 0;

        }
        System.out.println("A score of " + score+" on test "+testNumber+" gives a
        weighted score of "+weight +".");
        }
}
```

b) Write a method that solves the same problem described in part a) using only conditional expressions. The whole body of the method should be written using only one statement.

Write a `main` method that calls the method you implemented to display the following output for an input of 3 and 27:

```
A score of 27 on test 3 gives a weighted score of 5.4.
```

```
public class Score {

public static float score(int testNumber, float score) {

return (float) ((testNumber == 1) ? (0.1 * score) : (((testNumber == 2)
                || (testNumber == 3) || (testNumber == 6)) ? 0.2 * score
                : 0.15 * score));

}

public static void main(String[] args) {

        Scanner sc=new Scanner(System.in);
        System.out.println("please enter the test number and the score");
        int testNumber = sc.nextInt();
        int score = sc.nextInt();
        float weight = score(testNumber,score);
        System.out.println("A score of " + score+" on test "+testNumber+" gives
        a weighted score of "+weight+".");


}
```

Exercise 3 (8 Marks)

The method `Math.random()` gives a real number between 0.0 and 0.9999..., and so `6*Math.random()` is between 0.0 and 5.999.... The type-cast operator, `(int)`, can be used to convert this to an integer: `(int)(6*Math.random())`. Thus, `(int)(6*Math.random())` is one of the integers 0, 1, 2, 3, 4, and 5. To get a number between 1 and 6, we can add 1:

```
(int)(6*Math.random()) + 1
```

Using the statement above, we would like to know how many times we have to roll a pair of dice before they come up snake eyes? **Note:** Snake eyes means that both dice show a value of 1.

Write a method that should return the number of rolls that it makes before the pair of dice come up snake eyes.

The method should also display the following message, e.g.:

```
It took 100 rolls to get snake eyes.
```

Note: You have to use a do-while loop.

Solution:

```
public static void main(String[] args) {

     int die1, die2;   // The values rolled on the two dice.

     int countRolls;   // Used to count the number of rolls.

     countRolls = 0;

     do {
        die1 = (int)(Math.random()*6) + 1;   // roll the dice
        die2 = (int)(Math.random()*6) + 1;
        countRolls++;                        // and count this roll
     } while ( die1 != 1 || die2 != 1 );

     System.out.println("It took " + countRolls + " rolls to get snake eyes.");

 }
```

Exercise 4 (6+2+4=12 Marks)

Given the following code:

```
public static void main (String[] args) {

    int i, j, k;

    int x = 1;
    int y = 2;
    int z = 3;

    for (i=1; i<=x; i++) {
      for (j=1; j<=y; j++) {
        for (k=1; k<=z; k++) {
          System.out.println("i=" + i + ", j=" + j + ", k=" + k);
        }
      }
    }
}
```

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

Solution:

```
i=1, j=1, k=1
i=1, j=1, k=2
i=1, j=1, k=3
i=1, j=2, k=1
i=1, j=2, k=2
i=1, j=2, k=3
```

b) How many lines will be printed for any input `x`, `y`, and `z`?

Solution:

x*y*z

c) Change the code above to be able to print all permutations from 1 to `n`. For example for `n=3`, the following should be printed:

```
i=1, j=2, k=3
i=1, j=3, k=2
i=2, j=1, k=3
i=2, j=3, k=1
i=3, j=1, k=2
i=3, j=2, k=1
```

Solution:

```
public static void main (String[] args) {
    int i, j, k;
    int x = 3;
    int y = 3;
    int z = 3;
    for (i=1; i<=x; i++) {
      for (j=1; j<=y; j++) {
        for (k=1; k<=z; k++) {
            if(i!=j&&i!=k&&j!=k)
          System.out.println("i=" + i + ", j=" + j + ", k=" + k);
```

```
            }
          }
        }
    }
```

Exercise 5 (8+6=14 Marks)

a) Given a string `S` of length `n`, the Z-Algorithm produces a string `Z` where `Z.charAt(i)` is the length of the longest substring starting from `S.charAt(i)` which is also a prefix of `S`, i.e. the maximum `k` such that `S.charAt(j) == S.charAt(i+j)` for all $0 \leq j < k$. Note that `Z.charAt(i)=0` means that `S.charAt(0) != S.charAt(i)`.

The string `x` is a prefix of the string `w` if and only if `w = xy`.

Write a method that takes a string as argument and returns a string according to the Z-algorithm presented above.

Assume that `Z.charAt(0)` is always equal to `0`.

For example:

```
zFunction("ababa")     --> 00301
zFunction("axbyaxba")  --> 00003001
zFunction("abababab")  --> 006040200
zFunction("CSEN")      --> 0000
```

Explanation of the first sample run: `zFunction("ababa") --> 00301`

For the first character `a`, the z value is 0. For the second character `b`, the Z value is 0 since any prefix of `"ababa"` starts with `a`. For the third character `a`, the longest substring starting from the third position is `"aba"` that is also a prefix of `"ababa"`. Thus, the Z-value of `a` is 3. The fourth character `b` is having a Z-value 0. The last character `a` is having a Z-value 1.

Solution:

```
public static String z_AlgorithmA(String s) {
    String output = "0";
    for (int i = 1; i < s.length(); i++) {
        int count = 0;
        for (int j = 0; j < s.length(); j++) {
            if (((i + j)<s.length())&&s.charAt(j) == s.charAt(i + j))
                count++;
            else
                break;

        }
        output+=count;

    }
    return output;

}
```

b) The Z-Algorithm can be used to search a word `w` in a string `s`. Assuming that neither `w` nor `s` consists of the character `'%'`, if you concatenate `w` and `s` with a separator `%` and apply the Z-algorithm on the concatenated string, then the Z-algorithm will return a string that encodes important information about the occurrence of the word in the string.

Given the method above, write a method that takes two strings `w` and `s` and returns the number of occurrences of `w` in `s`.

For example:

```
occurences("ax", waxbyaxba")  --> 2
occurences("ax","abababababx")   --> 0
```

Assume that the longest substring at any index is of length `9`, therefore the Z-value does not exceed 9. To convert a character into a number, the method `Integer.parseInt()` method can be used as follows:

For example, for the character `'2'`, `Integer.parseInt('2' + "")` will return the integer `2`.

Solution:

```
public static int z_AlgorithmB(String w,String s)
    {
        int count=0;
        String concatenated=w+"%"+s;
        String z=z_AlgorithmA(concatenated);

        for(int i=0;i<z.length();i++)
        {
            int x= Integer.parseInt(z.charAt(i)+"");
            if(x==w.length())
                count++;

        }

        return count;
    }
```

Extra Sheet

Extra Sheet

Extra Sheet