Assignment 3

Topics: pseudocode and data conversion basics

1. In order to turn a String into an integer value you must use the method *Integer.parseInt(someParameter)*. For you to see how this actually works, try out this code yourself. Note that the code will not work right away. Please explain where it goes wrong and why? In order to run the code, comment out the part which does not work.

   ```
   String name = "Andrea";
   System.out.println("My name is " + name);
   int a = Integer.parseInt(name);        // here you turn the String name into an integer
   System.out.println("My variable a has value " + a);

   String ageString = "23";
   System.out.println("My age is " + ageString);
   int myAge = Integer.parseInt(ageString);     // here you turn the String ageString into
                                                // an integer
   int yourAge = 27;
   int totalAge = myAge + yourAge;
   System.out.println("Our combined ages amount to " + totalAge + " years");

   System.out.println("I am " + myAge + " years old while you are " + yourAge);
   System.out.println("I am " + ageString + " years old while you are " + yourAge);
   ```

   Notice that *myAge* is an *int* and *ageString* is a *String*. Yet, they print the same or…? Please, try to ponder about this. Also make a few of your own tests, playing around with *int* and String but also with *float* and *String*. To help you in this latter case, you should know that there is a method that, given a *String* returns a *float*, whenever possible. It can be used in this way

   ```
   aFloat = Float.parseFloat(aString)
   ```

   Try to generalize and see how and which elementary data types can be created out of a *String* variable. Other said, try to see if there are more *parseXXX(aString)* methods and look what they return.

2. Write the pseudocode (i.e. a description of the steps necessary to get to a solution of the problem) for a software application that takes two values as input, check that the values passed are two and that they are both integer values. If they are not, simply exit the application. Otherwise, given the two numbers N and M print out row by row the

multiplication table of size N rows, M columns. For instance, for N = 3 and M = 5 you will have to print out:

1 2 3 4 5
2 4 6 8 10
3 6 9 12 15

3. Write the pseudocode (i.e. a description of the steps necessary to get to a solution of the problem) for a software application that accepts an integer N as input from the user. Then check if the number is a positive integer. If it is negative, just drop the sign and make it a positive number. If it is not an integer, exit the application.
   Given a positive number instead, print out all the prime numbers up to the number entered. For instance, given -32 as input your application will have to display these numbers:

   2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31

   Note: a number is a prime if it cannot be divided by any of its predecessors.

4. Write the pseudocode (i.e. a description of the steps necessary to get to a solution of the problem) for a software application that simulates the lotto draw. A lotto draw occurs as following: a random number between 0 and 90 is selected and displayed, then just wait, say two seconds, before displaying the next random number. The same number cannot occur later again. Wait again, and then proceed as before. The total numbers to be selected are 6. Add what is called jolly number i.e. once the 6 random numbers have been selected, another number, that can also be the same as one of those drawn before, is displayed (it should be preceded by some special markers to highlight it is a jolly). This is an example of a possible output:

   4 53 21 7 68 ---- 53

5. Write the pseudocode (i.e. a description of the steps necessary to get to a solution of the problem) for a software application that generates random names. A random name is a sequence of characters of a given length. Be the length also random chosen within some predefined range (e.g. between 2 and 12 letters). Only names that comply with these rules are accepted:

   - The name cannot have occurrences of more than three sequential consonants (e.g. a name like a**ndr**ea is ok, but a name like a**ndrt**ea is not)
   - The name cannot contain triples i.e. a**nnn**a is not a name
   - The name can contain only letters from a to z, and the additional letters å,æ,ø only