Object Oriented Programming TICS-201 (Viña-Stgo)

Professor: Ricardo Seguel, Ph.D.

29-Mar-2018

UNIVERSIDAD ADOLFO IBÁÑEZ

Test 2 (30 min)

Professor: Ricardo Seguel, Ph.D.

29-Mar-2018

Control 2 - (30 min)

- Programe en **Eclipse** la Clase Java ***Gambler*** (apostador) que recibe como argumentos:
  - capital inicial (***stake***), capital objetivo (***goal***) e intentos (***trials***)
- Un intento consiste en calcular un acierto o falla como si se tirara una moneda. Recuerde la función **random** vista en clases.
- Si acierta se suma 1 a ***stake***
- Si falla se resta 1 a ***stake***
- El apostador para si realiza todos los intentos (***trials***) y cuenta los intentos en los que ganó alcanzando el objetivo (***stake*** = ***goal***) y cuenta los intentos en los que perdió (***stake*** = 0)
- La función debe retornar en pantalla la cantidad de intentos en que ganó y perdió dentro del total de ensayos (***trials***)
- Suba su progrma al link habilitado en Webcursos.
- Su programa debe compilar y ejecutar sin errores retornando en la consola de **Eclipse,** por ejemplo con los argumentos 5 25 1000 imprime algo como esto:

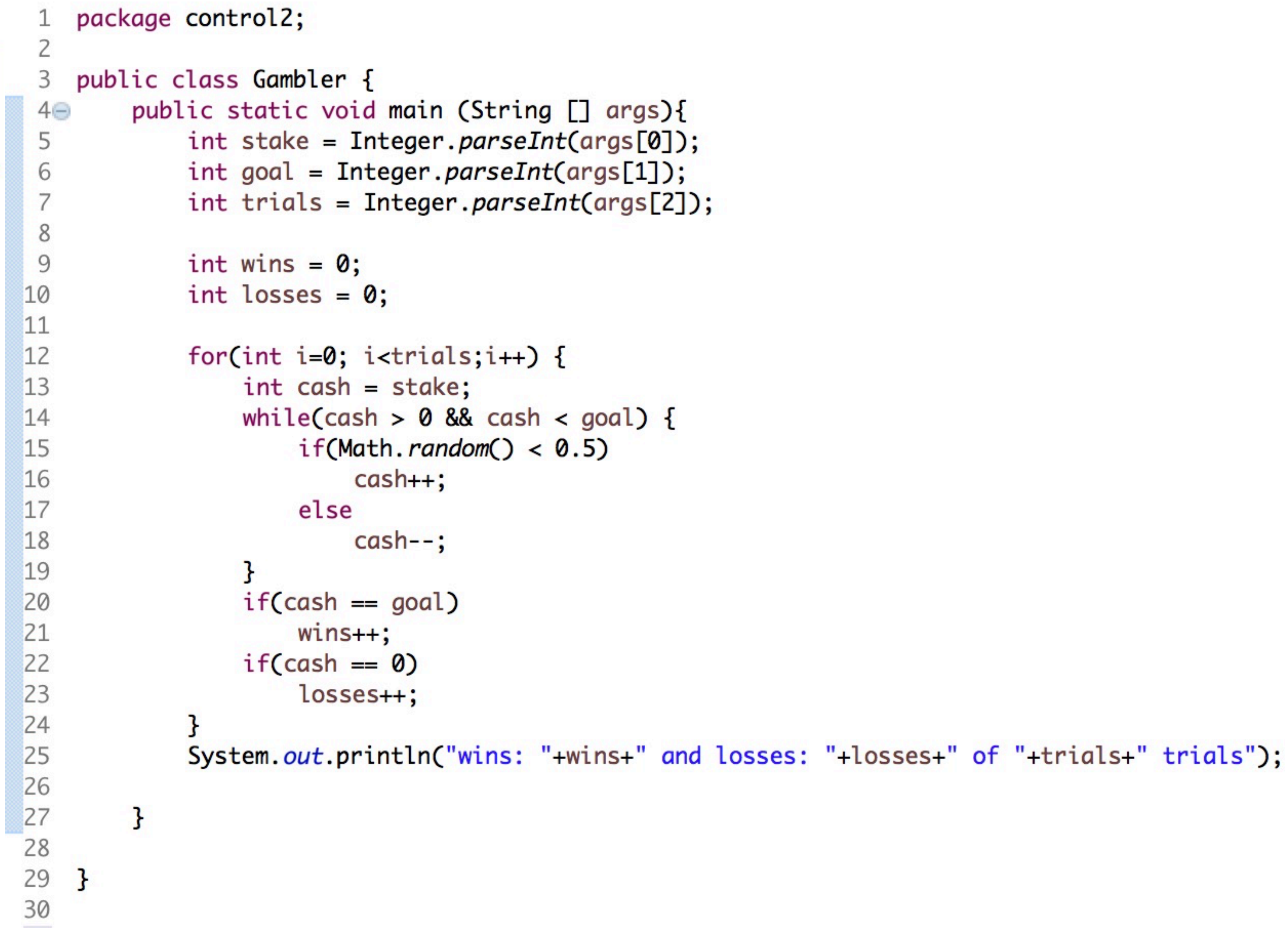
```
wins: 195 and losses: 805 of 1000 trials
```

UNIVERSIDAD ADOLFO IBÁÑEZ

Answers

Professor: Ricardo Seguel, Ph.D.

29-Mar-2018



```java
package control2;

public class Gambler {
    public static void main (String [] args){
        int stake = Integer.parseInt(args[0]);
        int goal = Integer.parseInt(args[1]);
        int trials = Integer.parseInt(args[2]);

        int wins = 0;
        int losses = 0;

        for(int i=0; i<trials;i++) {
            int cash = stake;
            while(cash > 0 && cash < goal) {
                if(Math.random() < 0.5)
                    cash++;
                else
                    cash--;
            }
            if(cash == goal)
                wins++;
            if(cash == 0)
                losses++;
        }
        System.out.println("wins: "+wins+" and losses: "+losses+" of "+trials+" trials");

    }

}

```

UNIVERSIDAD ADOLFO IBÁÑEZ

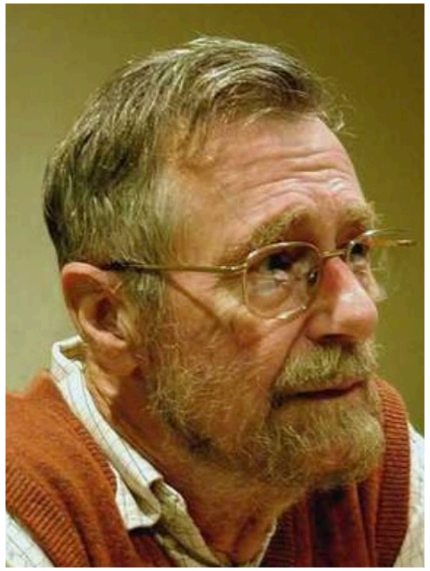
Ok, let’s continue…

“The question of whether computers can think is like the question of whether submarines can swim. ”

“Program testing can be used to show the presence of bugs, but never to show their absence!”

“Too few people recognize that the high technology so celebrated today is ***essentially a mathematical technology***.”

Edsger W. Dijkstra



What we have studied last week

- Key basic concepts of Programming
- Why Java
- Key concepts of Java
  - Data types
  - Operators
  - Strings
  - Loops
  - Conditions

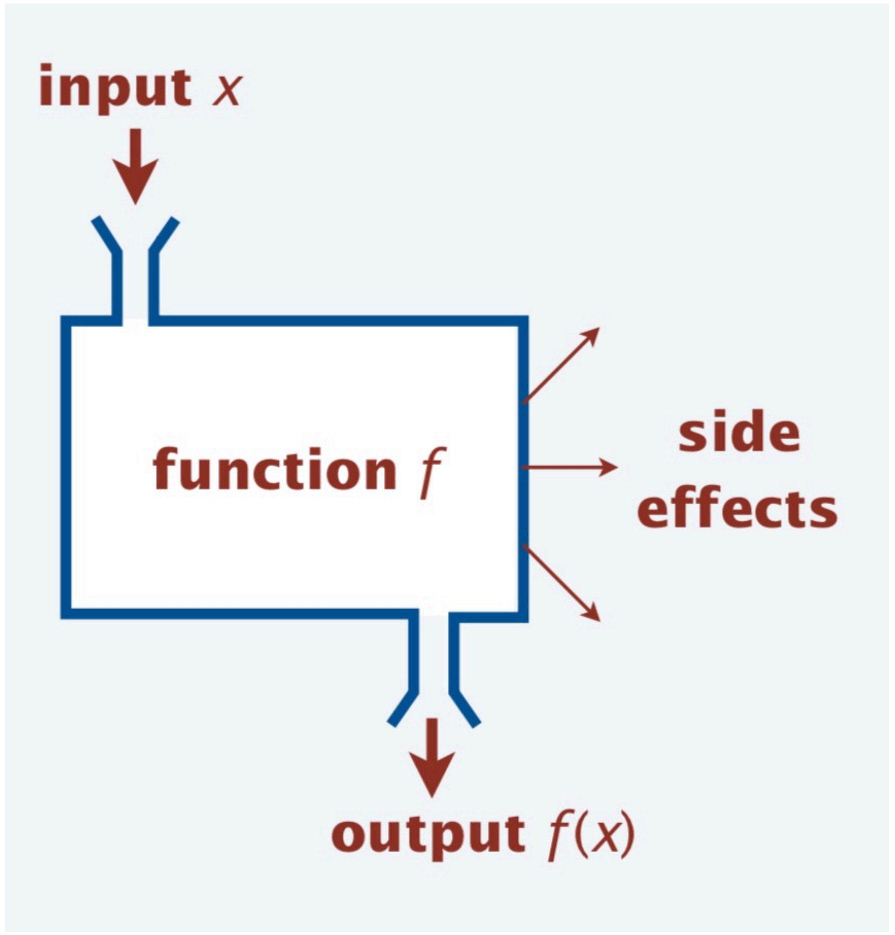
What we’ll study today

- Functions
- Hands-on of the exercises of Chapters 3 and 4 of the guide Java Book of the course

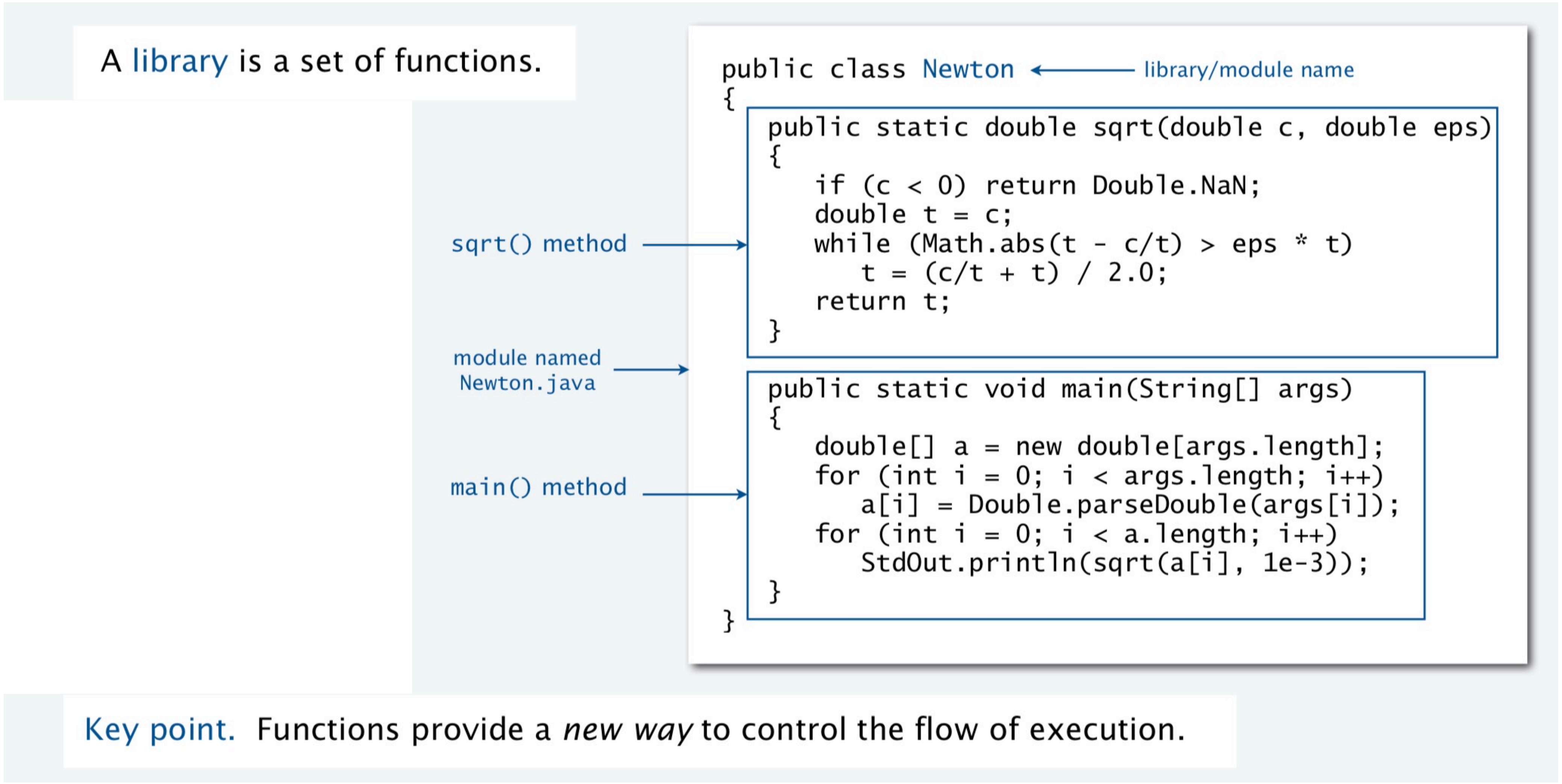
Functions

Applications
- Scientists use mathematical functions to calculate formulas.
- Programmers use functions to build modular programs.
- You use functions for both.

Examples seen so far
- Built-in functions: Math.random(), Math.abs(), Integer.parseInt().
- User-defined functions: main().



Anatomy of a Java Library



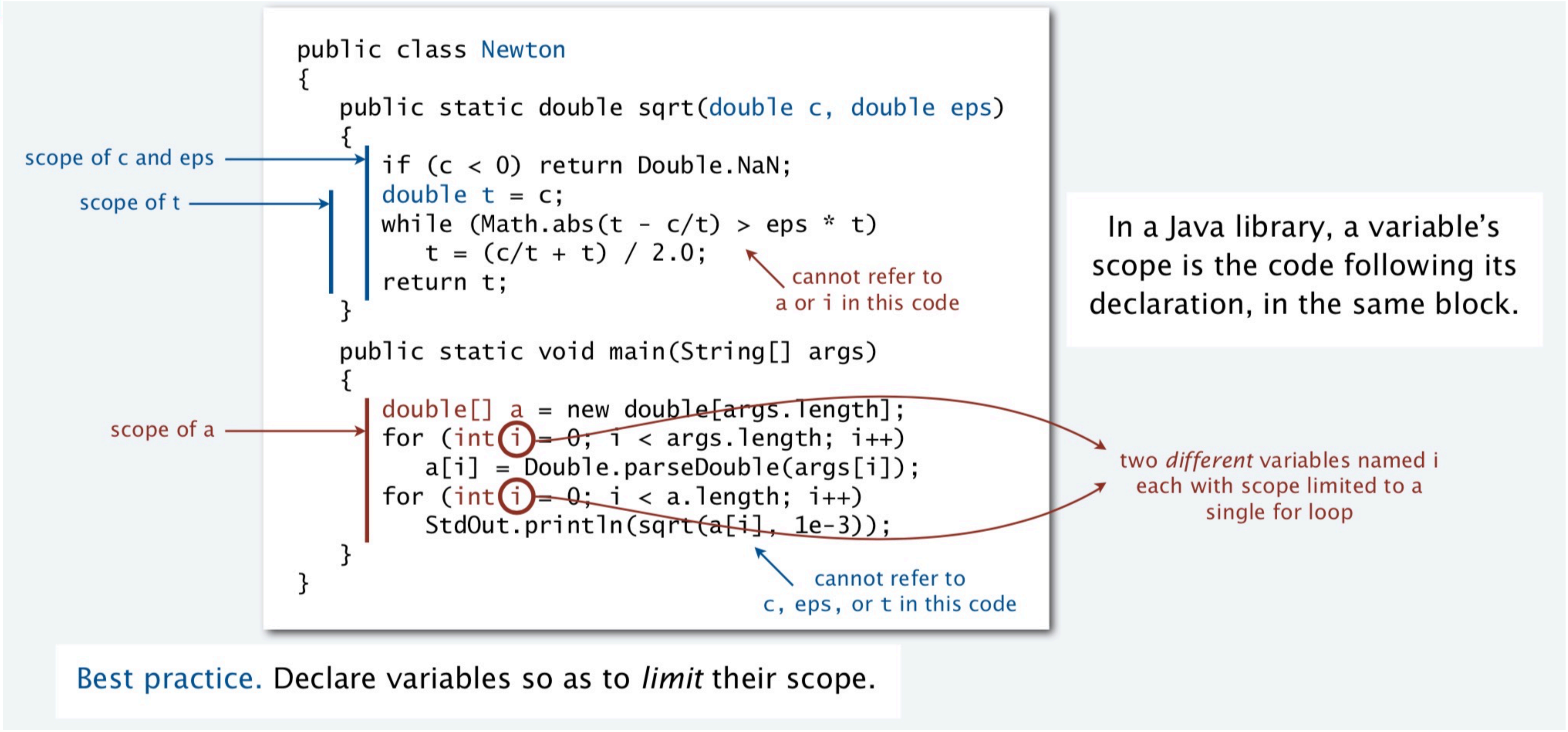
A library is a set of functions.

```
public class Newton
{
   public static double sqrt(double c, double eps)
   {
      if (c < 0) return Double.NaN;
      double t = c;
      while (Math.abs(t - c/t) > eps * t)
         t = (c/t + t) / 2.0;
      return t;
   }

   public static void main(String[] args)
   {
      double[] a = new double[args.length];
      for (int i = 0; i < args.length; i++)
         a[i] = Double.parseDouble(args[i]);
      for (int i = 0; i < a.length; i++)
         StdOut.println(sqrt(a[i], 1e-3));
   }
}
```

Key point. Functions provide a *new way* to control the flow of execution.

Scope

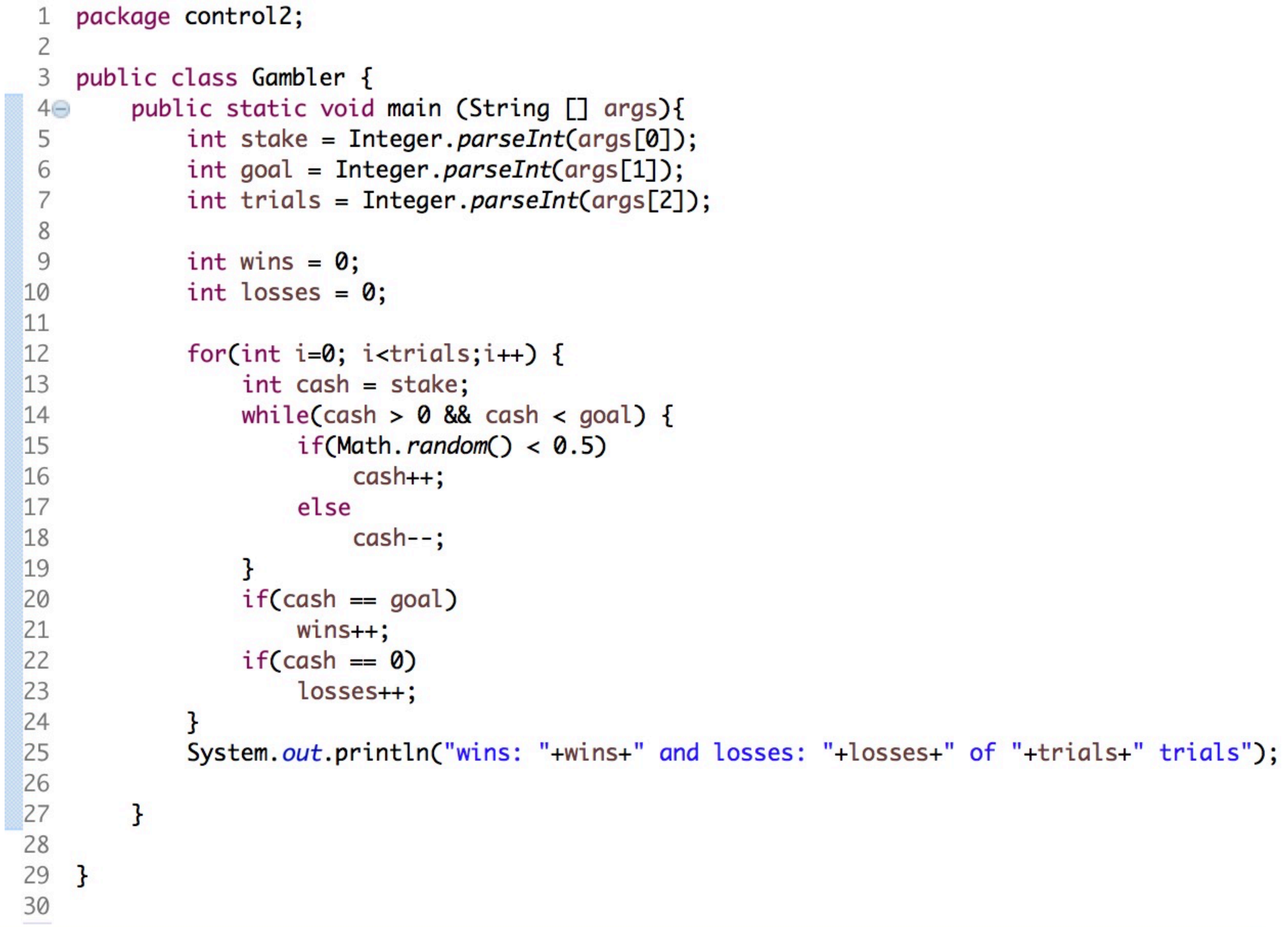

```java
public class Newton
{
   public static double sqrt(double c, double eps)
   {
      if (c < 0) return Double.NaN;
      double t = c;
      while (Math.abs(t - c/t) > eps * t)
         t = (c/t + t) / 2.0;
      return t;
   }

   public static void main(String[] args)
   {
      double[] a = new double[args.length];
      for (int i = 0; i < args.length; i++)
         a[i] = Double.parseDouble(args[i]);
      for (int i = 0; i < a.length; i++)
         StdOut.println(sqrt(a[i], 1e-3));
   }
}
```
scope of c and eps
scope of t
cannot refer to a or i in this code
scope of a
two *different* variables named i each with scope limited to a single for loop
cannot refer to c, eps, or t in this code

In a Java library, a variable's scope is the code following its declaration, in the same block.

Best practice. Declare variables so as to *limit* their scope.

Change the program to use a method called Trial(s,g)



```java
package control2;

public class Gambler {
    public static void main (String [] args){
        int stake = Integer.parseInt(args[0]);
        int goal = Integer.parseInt(args[1]);
        int trials = Integer.parseInt(args[2]);

        int wins = 0;
        int losses = 0;

        for(int i=0; i<trials;i++) {
            int cash = stake;
            while(cash > 0 && cash < goal) {
                if(Math.random() < 0.5)
                    cash++;
                else
                    cash--;
            }
            if(cash == goal)
                wins++;
            if(cash == 0)
                losses++;
        }
        System.out.println("wins: "+wins+" and losses: "+losses+" of "+trials+" trials");

    }

}

```

Showtime!

UNIVERSIDAD ADOLFO IBÁÑEZ

Show your homework (Exercise 3)

Exercise 3:

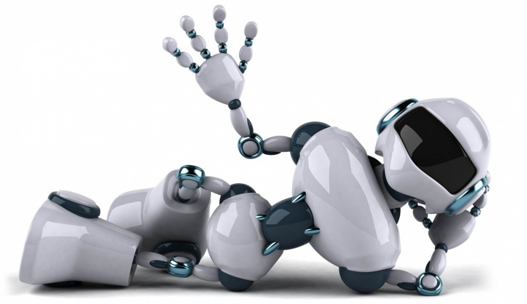
- Code a Java program that receives as input a date as `name dd mm yy`
- Take the input and produce the output date in the American format and the European format

```
American format:
Thursday, July 16, 2015
European format:
Thursday 16 July 2015
```

- Add the conditions to select the format as an argument
- Add the loop to show all the days after the received date until the last day of the month

Show your project advance for this week

Course Project



- Start transforming your Assistant into Java code
- Every week assignment. Small chunks -> Agile!
- We will have weekly advances as assignments for demonstrations in the class
- General Requirement:
  - Code an automated assistant to keep you updated with your to-do list and give you alerts for every task and inform you about important news and calls you received during your busy time (performing a task).
- Specific requirements for the next week to start coding with Java
  - Use console input/output (not a file yet, if so it's a plus)
  - Use conditions and loops
  - Follow the instructions on how to write clear code published in Webcursos

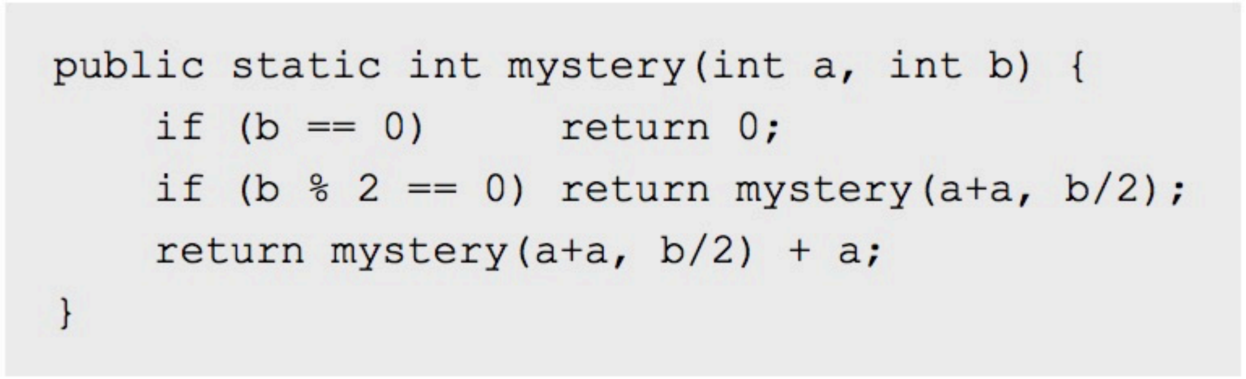
UNIVERSIDAD ADOLFO IBÁÑEZ

Hands-on

Coding Java in Eclipse

(Follow the instructions on how to write clear code published in Webcursos)

Exercise 4

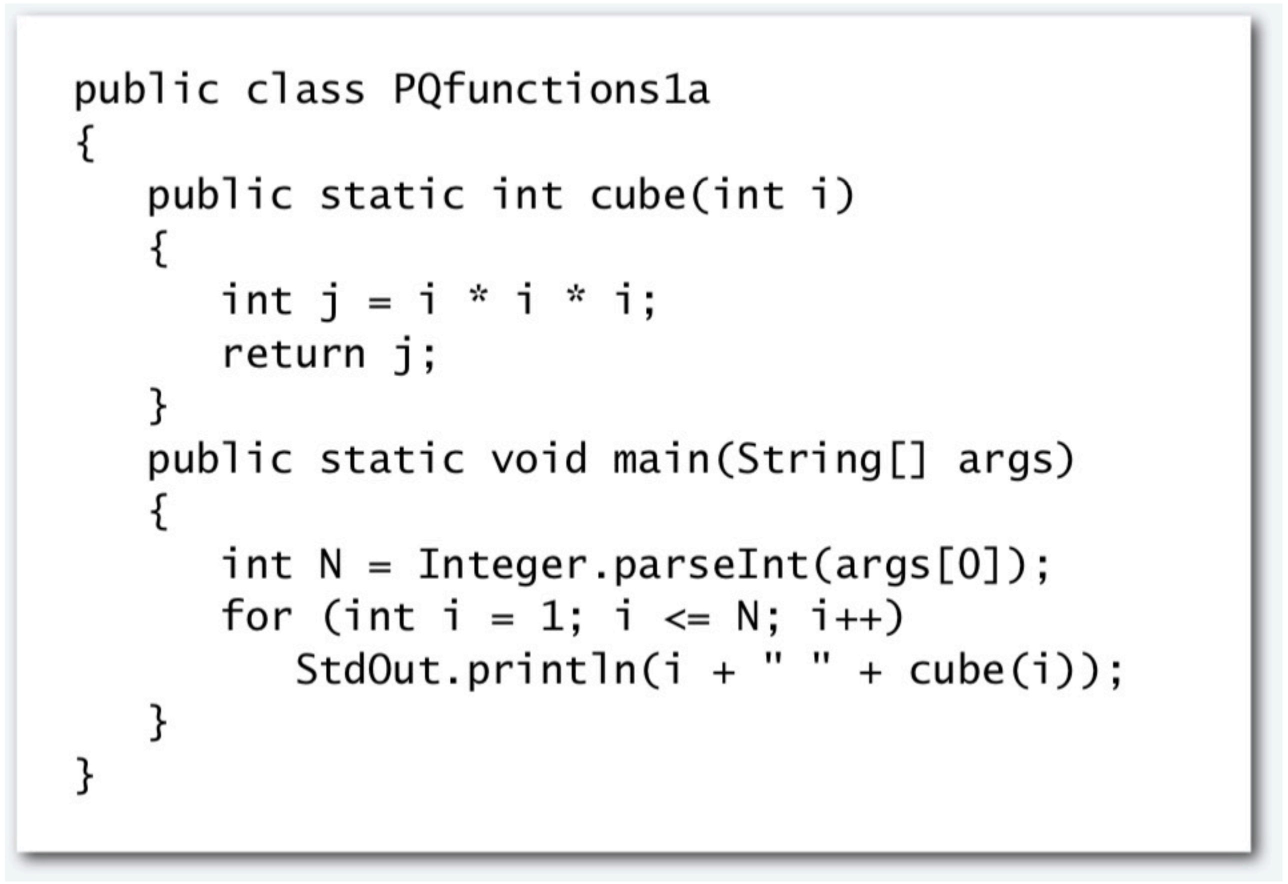


```
public static int mystery(int a, int b) {
    if (b == 0)      return 0;
    if (b % 2 == 0) return mystery(a+a, b/2);
    return mystery(a+a, b/2) + a;
}
```

- Program and run the function ***mystery***
- What are the values of mystery(2, 25) and mystery(3, 11)?
- Given positive integers a and b, describe what value mystery(a, b) computes.
- Answer the same question, but replace + with * and replace return 0 with return 1.

Exercise 5

- What happens when you compile and run the following code?



```java
public class PQfunctions1a
{
   public static int cube(int i)
   {
      int j = i * i * i;
      return j;
   }
   public static void main(String[] args)
   {
      int N = Integer.parseInt(args[0]);
      for (int i = 1; i <= N; i++)
         StdOut.println(i + " " + cube(i));
   }
}
```

UNIVERSIDAD ADOLFO IBÁÑEZ

Exercise 6 (from Chapter 3)

- Program and run the solution for the Exercise 3.4 of the guide book (Think Java)

Exercise 7 (from Chapter 4)

- Solve, program and run Exercise 4.1 (points 1, 2 and 3) of the guide book (Think Java)

Exercise 8 (from Chapter 4)

- Program and run Exercise 4.3 of the guide book (Think Java)

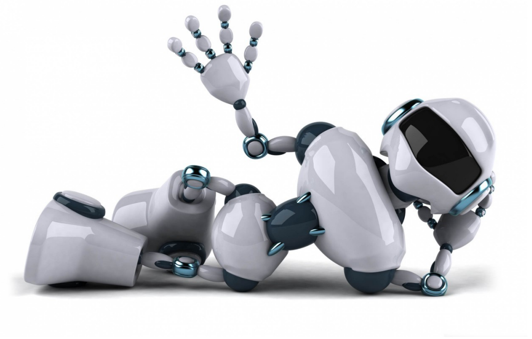
UNIVERSIDAD ADOLFO IBÁÑEZ

Next Week…

Compulsory Self study for next week

- **Compulsory** reading of Chapter 5 and 6 of the Book "Think Java: How to Think Like a Computer Scientist" for preparing your self for the next lecture and the test of the week after.
- **Compulsory** reading of Complementary articles (this material will be part of the questions in **P1**)
  - *Secure Software Development Life Cycle Processes*
- **Test 3** covering chapters 1-4 guide book (Think Java) + all Java lessons
- Make the appointment in your agenda:
  - **P1 on April 19 2018 at 11.30**
  - Selection of alternatives
  - Java Programming with Eclipse

Course Project



- Every week assignment. Small chunks -> Agile!
- We will have weekly advances as assignments for demonstrations in the class
- General Requirement:
  - Code an automated assistant to keep you updated with your to-do list and give you alerts for every task and inform you about important news and calls you received during your busy time (performing a task).
- Specific requirements <u>for the next week </u>to continue coding with Java
  - Use console input/output (not a file yet, if so it's a plus)
  - Use conditions, loops, and void or value methods
  - Follow the instructions on how to write clear code published in Webcursos

Homework

- Practice and finish the exercises 6, 7 and 8 to show them the next week in the class
- Submit your solution to the available link by next Thursday April 5 10.00 am, before the class

UNIVERSIDAD ADOLFO IBÁÑEZ

Object Oriented Programming TICS-201 (Viña-Stgo)

Professor: Ricardo Seguel, Ph.D.

29-Mar-2018