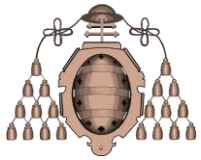
Unit 2

Introduction to programming

Computing Basics



University of Oviedo

Table of contents

- 2.1 Problem abstraction for programming. Basic concepts.
- 2.2 Variables, expressions, assignment
- 2.3 Console input / output
- 2.4 Basic structures for control flow handling: sequential, choice and repetitive
- 2.5 Definition and use of subprograms and functions. Variable scope
- 2.6 File input / output
- 2.7 Basic data types and structures: arrays

What is a program?

- The main reason for writing programs is to be able to **solve problems** using a computer.
- A **program** is a text that contains a sequence of instructions suitable enough to be interpreted and executed by computers.

Python

```
print("Hello world")
```

```
#include <stdio.h>

int main()
{
  printf ("Hello world");
  return 0;
}
```

C

```
# Program that solves the problem
# of computing the surface of a
# triangle
base = float(input("Type the
base: "))
height = float(input("Type the
height: "))

surface = (base*height)/2
print("The surface is: ",
surface)
```

Python

How a program is written?

- Before starting to write a program, it is necessary to analyze the problem to be solved.
- It is **indispensable** to identify the required steps to solve the problem.
- An **algorithm** is an unambiguous, finite and orderly sequence of instructions that must be followed in order to solve a problem.

- Ask for the base and height
- Compute the surface
- Show the result

Algorithm to compute the surface of a triangle

How a program is written?

- A **program** is an algorithm written in a certain programming language.

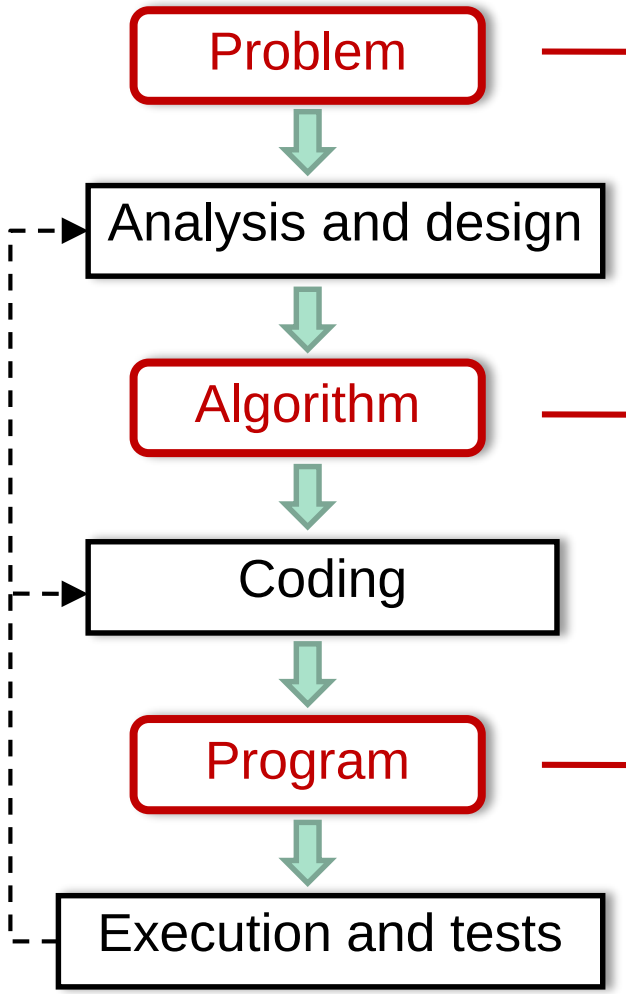
  - Ask for the base and height
  - Compute the surface
  - Show the result

```python
#Program that solves the problem of
#computing the surface of a triangle
base = float(input("Type the base: "))
height = float(input("Type the height: "))

surface = (base*height)/2
print("The surface is: ", surface)
```

Python

Program design process



Natural language

```
We need to compute the
surface of a triangle
```

Natural language
pseudocode

```
1.Ask for the base and
  height
2.Compute the surface
3.Show the result
```

Programming
language

Python

```
base=float(input("Base:"))
height=float(input("Height:"))

surface=(base*height)/2

print("Surface is: ", surface)
```

Programming languages

- A **programming language** is composed of a set of symbols and syntactic and semantic rules, which define both the structure and the meaning of the instructions that appear in the source code.

Some programming languages:

Basic	C/C++	Perl
Pascal	Java	**Python**

- Examples:

```
print("Hello world")
```
Python

```
printf("Hello world");
```
C

```
System.out.print("Hello world");
```
Java

```
cout << "Hello world";
```
C++

Programming paradigms

Some programming paradigms:

- **Modular**
- Objected oriented
- Logic
- Functional
- Imperative (using procedures)
- Declarative

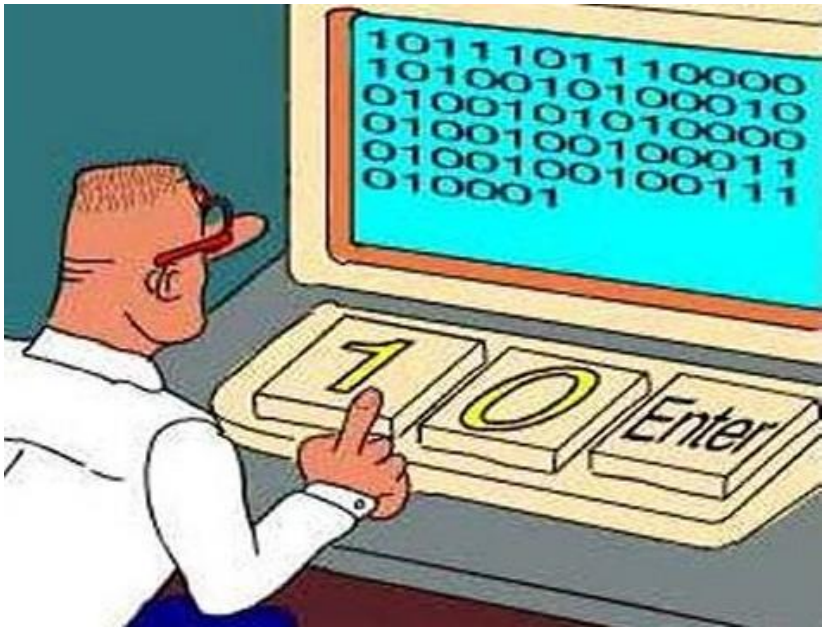
- **Modular programming** is a programming technique that consists in dividing a program into modules or subprograms in order to make it more readable and easier to handle.

High and low level languages

- Human beings program using **high-level languages.**
- Computers only understand binary / machine language.

Programs written in a high-level language have to be **translated** into machine language:

- Compiled languages.
- Interpreted languages.



Compilers and interpreters

- **Compiled languages:** The *compiler* performs a complete translation of the program written in a high-level language into its machine language counterpart (executable).

- **Interpreted languages:** The *interpreter* reads and executes the high-level program line by line.