CC - AI

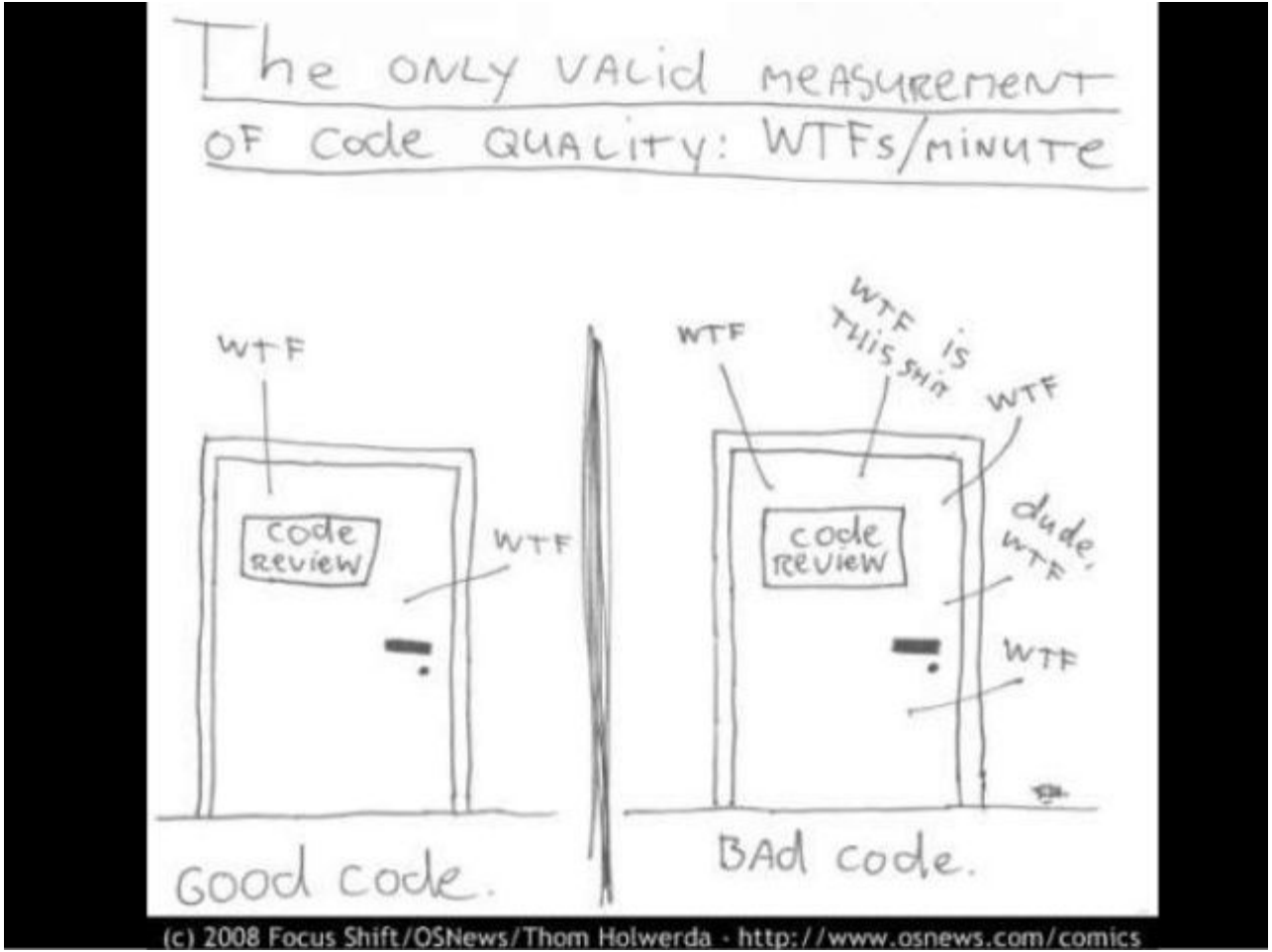
Programación para la Computación Científica - IA

Programming Methodology

Universidad Sergio Arboleda
Prof. John Corredor

Today goal's

- To understand Honor Code
- To become familiar with effective programming practices that result in the development of correct programs with minimum effort
- To become familiar with Program Documentation


The ONLY VALID MEASUREMENT OF CODE QUALITY: WTFs/MINUTE
WTF
code review
WTF
Good code.
WTF
WTF is THIS SHIT
WTF
code review
dude, WTF
WTF
BAD code.
(c) 2008 Focus Shift/OSNews/Thom Holwerda - http://www.osnews.com/comics

Programming Methodology?

- The process used by an individual or a team for developing programs

Good Programming Methodology?

- A methodology that enables the lowest-cost and on-schedule development of programs that are correct, easy to maintain & enhance

Correct program?

- A program with correct syntax & semantics

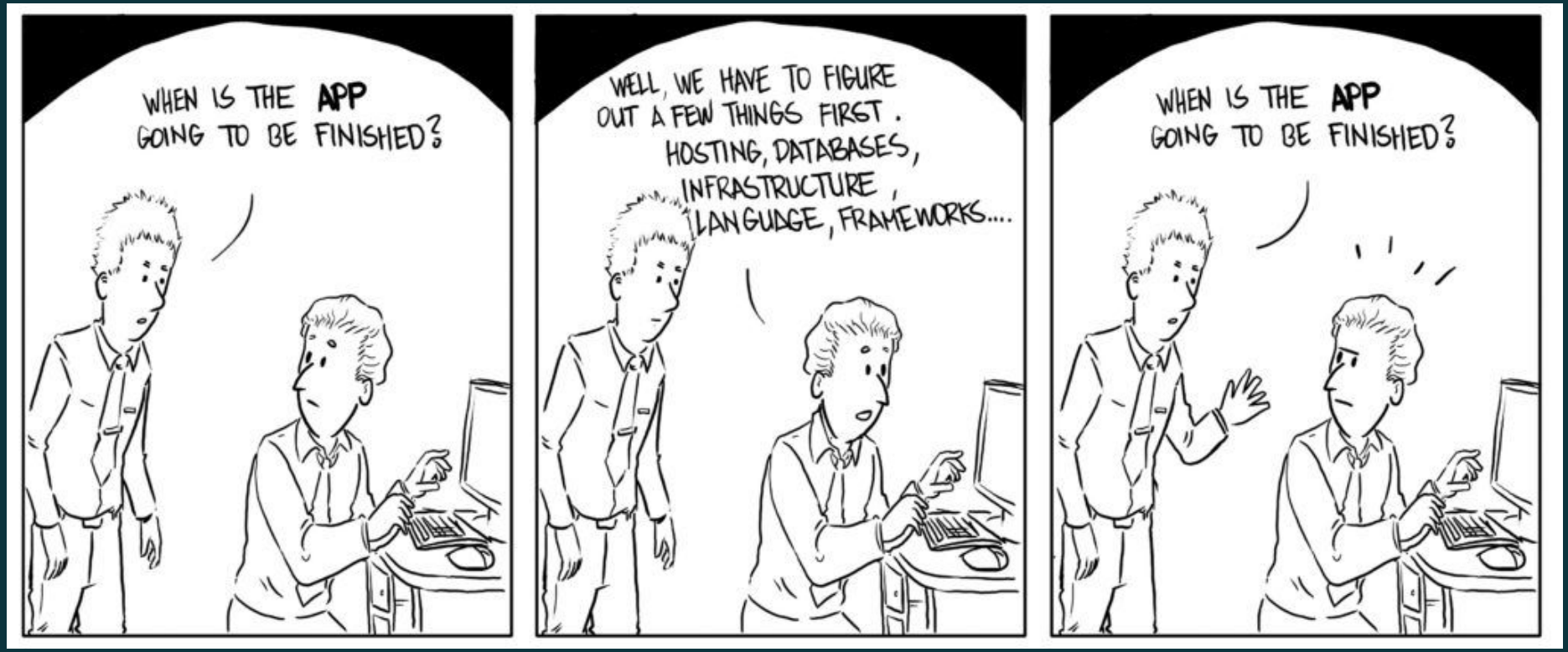
Readable program?

- A program that is easy to read & understand, and therefore, easy to maintain & enhance

Introduction

When programs are developed to solve real-life problems like inventory management, payroll processing, student admissions, examination result processing, etc. they tend to be huge and complex.

The approach to analyzing such complex problems, planning for software development and controlling the development process is called programming methodology.


WHEN IS THE APP GOING TO BE FINISHED?
WELL, WE HAVE TO FIGURE OUT A FEW THINGS FIRST.
HOSTING, DATABASES, INFRASTRUCTURE, LANGUAGE, FRAMEWORKS....
WHEN IS THE APP GOING TO BE FINISHED?

Procedural Programming

Problem is broken down into procedures, or blocks of code that perform one task each. All procedures taken together form the whole program. It is suitable only for small programs that have low level of complexity.

Example

For a calculator program that does addition, subtraction, multiplication, division, square root and comparison, each of these operations can be developed as separate procedures. In the main program each procedure would be invoked on the basis of user's choice.

```
# Program make a simple calculator
# This function adds two numbers
def add(x,  y):
   return x + y
print("Select operation:   ")
print("1.Add")
print("2.Subtract")
# Take input from th
choice = input("En
num1 = float(input(
if choice == '1':
   print(num1,"+",n
elif choice == '2':
   print(num1,"-",nu
else:
   print("Invalid input")
```

Challenge 01!!!

```
corredor@john:~/Documents/A-Clases/A-Cientific
Select operation.
1.Add
2.Subtract
3.Multiply
4.Divide
Enter choice( 1 | 2 | 3 | 4 ): 4
Enter first number: 100
Enter second number: 3
100.0 / 3.0 = 33.333333333333336
corredor@john:~/Documents/A-Clases/A-Cientific
```

Object-Oriented Programming

Here the solution revolves around entities or objects that are part of problem. The solution deals with how to store data related to the entities, how the entities behave and how they interact with each other to give a cohesive solution.

Example

If we have to develop a payroll management system, we will have entities like employees, salary structure, leave rules, etc. around which the solution must be built.

```
class Dog:
    # Class Attribute
    species = 'mammal'
    # Initializer / Instance Attributes
    def __init__(self, name, age):
        self.name = name
        self
    # insta
    def de
        retu
    # insta
    def sp
        return "{} says {}".format(self.name, sound)
# Instantiate the Dog object
rocco = Dog("Rocco Daniel", 4)
# call our instance methods
print(rocco.description())
```

Challenge 02!!!

```
corredor@john:~/Documents/A-Clases/A-Cientifica/py
Rocco Daniel is 3 years old
Rocco Daniel says Grrrrrr Gruff Gruff
corredor@john:~/Documents/A-Clases/A-Cientifica/py
```

Programming Methodology

- Programming is all about solving a particular problem through computerized codes.
- Whether it be a problem of inventory management, running of a remote car, or even running of a missile, among others.
- Thus the scope of programming ranges from **very simpleton** tasks to **extremely complicated** ones.
- But behind all the codes, one parameter is common, that is, to handle and **solve the problem efficiently**.

Programming Methodology

- As one can’t learn to fly an aeroplane simply by watching it flying, one has to actually learn it.

- Similarly to become a proficient coder, one has to actually do the codes. And, the task of coding is extremely easy once one learns how to apply the **logic for problem solving**.

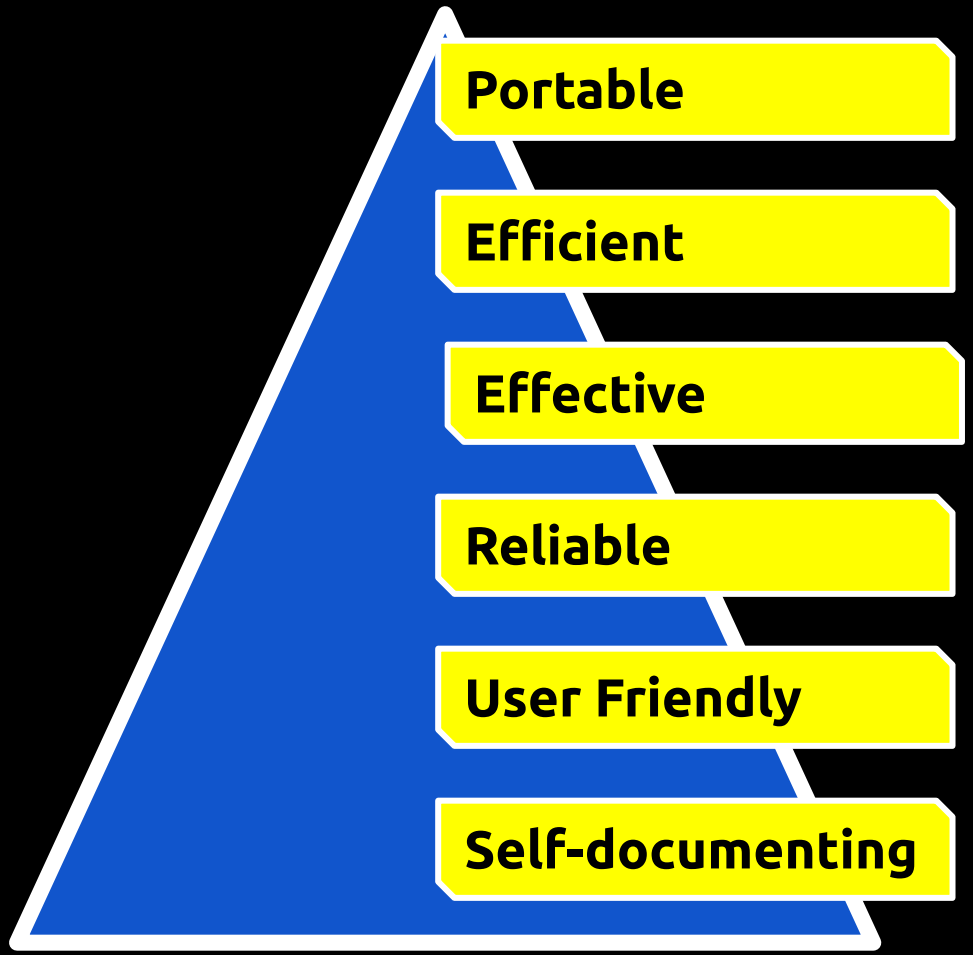
Facing the problem

A typical software development process (steps) –

- Requirement gathering
- Problem definition
- System design
- Implemen
- Testing
- Document
- Training a
- Maintenan

The first two steps assist the team in understanding the problem, the most crucial first step towards getting a solution. Person responsible for gathering requirement, defining the problem and designing the system is called **system analyst**.

Characteristics of a Program



Any program or software whose identifier names, module names, etc. can describe itself due to use of explicit names.

Proper Identifier Names

A name that identifies
variable, object, functi
class or method is call
identifier. Giving prop
identifier names make
program self-documen
This means that name
object will tell what it
or what information it

- Use language guidelines
- Don’t shy from giving long names to maintain clarity
- Use uppercase and lowercase letters
- Don’t give same name to two identifiers even if the language allows it
- Don’t give same names to more than one identifier even if they have mutually exclusive scope

Comments

It tells the reader that the next few lines of code will give information about....

This line is not part of the code but given only to make the program more user friendly.

Comments can be inserted as –

- Prologue to the program to explain its objective
- At the beginning and/or end of logical or functional blocks
- Make note about special scenarios or exceptions

You should avoid adding superfluous comments as that may prove counterproductive by breaking the flow of code while reading. Compiler may ignore comments and indentations but the reader tends to read each one of them.

Program Documentation

- Any written text, illustrations or video that describe a software or program to its users is called program or software document.
- User can be anyone from a programmer, system analyst and administrator to end user.
- At various stages of development multiple documents may be created for different users.
- In fact, software documentation is a **critical process** in the overall software development process.

These are some guidelines for creating the documents –

- Documentation should be from the point of view of the reader
- Document should be unambiguous
- There should be no repetition
- Industry standards should be used
- Documents should always be updated
- Any outdated document should be phased out after due recording of the phase out

Advantages of Documentation

- Keeps track of all parts of a software or program
- Maintenance is easier
- Programmers other than the developer can understand all aspects of software
- Improves overall quality of the software
- Assists in user training
- Ensures knowledge de-centralization, cutting costs and effort if people leave the system abruptly

References

- Kernighan, Brian W., and Dennis M. Ritchie. The C Programming Language. Vol. 2. Englewood Cliffs: prentice-Hall, 1988.
- Silberschatz, Abraham, Peter B. Galvin, and Greg Gagne. Operating System Concepts. Vol. 8. Wiley, 2013.
- https://www.w3schools.com/python/python_intro.asp
- https://www.programiz.com/python-programming/methods/dictionary
- https://www.tutorialspoint.com/programming_methodologies/programming_methodologies_program_documentation.htm