

Programming

8 + 3 = 11 h./week

Programming

<http://www.dreamfoundation.eu/english-test>

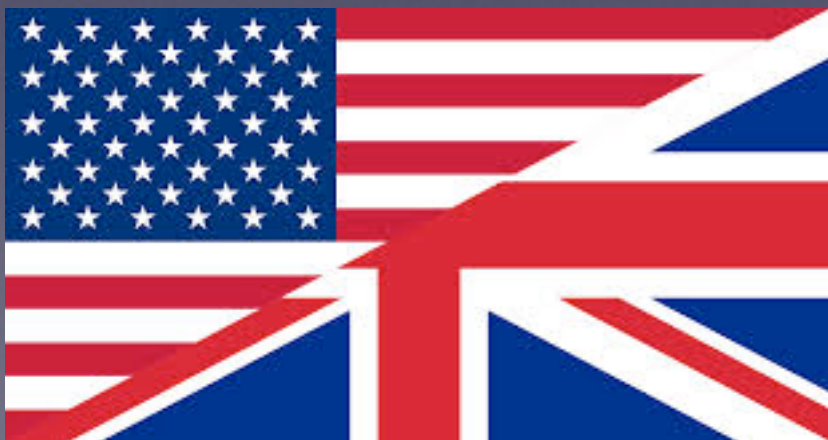
Programming

8 + 3 = 11 h./week



+

=



Programming



English in real life

Programming

You are not prefect

I am not prefect

Programming



Relaxing cup of café con leche

Evaluation

- 80 % Exam + compulsory works
- 20 % attitudes

Evaluation

- 80 % Exam + compulsory works
- 20 % attitudes (punctuality, work in group, respect)

Evaluation

- 80 % Exam + compulsory works
- 20 % attitudes (punctuality, work in group, respect)



5% use of English

Evaluation

Remedial exam of every evaluation

After the 3^d eval.

Evaluation

Remedial exam of every evaluation

After the 3^d eval.

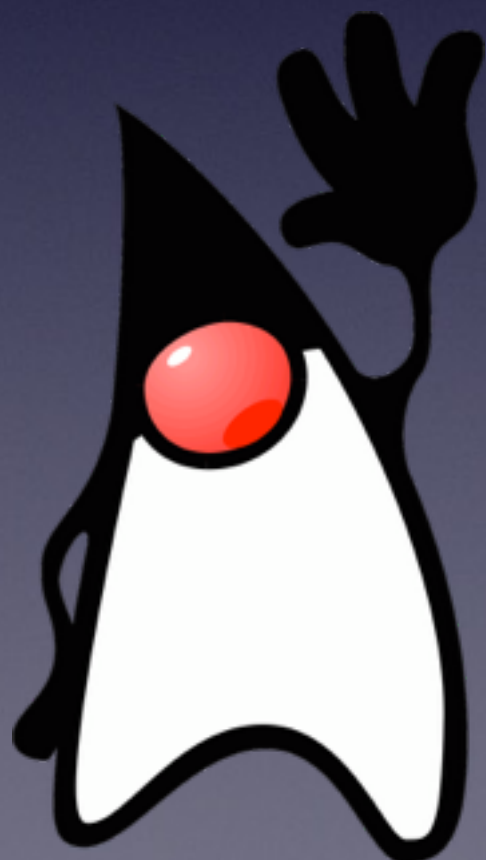
Remedial September exam (ALL)



Java™



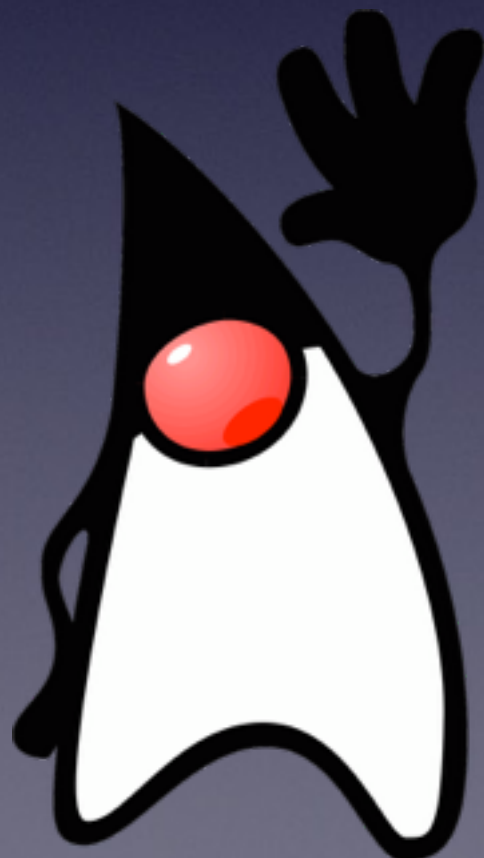
Java™



Duke



Java™



fp.edu.gva.es

User: 0+DNI

Password: 0+DNI

Programming

Unit I. Introduction



I. Sum, subtract, multiply or divide 2 numbers



1. Sum, subtract, multiply or divide 2 numbers
2. Compare 2 values



1. Sum, subtract, multiply or divide 2 numbers
2. Compare 2 values
3. Store and retrieve information



Easy operations ... but ...



Easy operations ... but ...



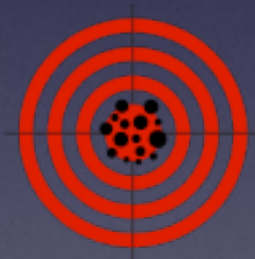
very fast



Easy operations ... but ...



very fast



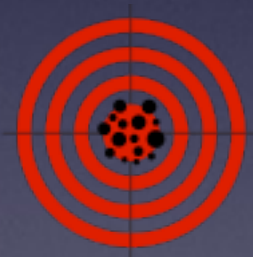
reliable



Easy operations ... but ...



very fast



reliable



big capacity

Algorithm / Program

Algorithm

Algorithm: is a step-by-step **procedure** to solve a problem in a **finite amount of time**.



Algorithm / Program

Algorithm

Algorithm: is a step-by-step **procedure** to solve a problem in a **finite amount of time.**



Procedure: is a **sequence** of instructions that can be performed in a **mechanical** way.

Algorithm / Program

Algorithm

15 coins game.

This two-players game starts placing 15 coins on the table.

Player one picks from 1 to 3 coins. Next, player 2 picks from 1 to 3 coins from the remaining coins on the table. This process is repeated until there are no coins left on the table. The player which is forced to take the last coin loses.

The problem consists on finding a winning strategy that allows player 1 to force player 2 to take the last coin.

Algorithm / Program

Algorithm

15 coins game.

This two-players game starts placing 15 coins on the table.

Player one picks from 1 to 3 coins. Next, player 2 picks from 1 to 3 coins from the remaining coins on the table. This process is repeated until there are no coins left on the table. The player which is forced to take the last coin loses.

The problem consists on finding a winning strategy that allows player 1 to force player 2 to take the last coin.

- 1.- Player 1 takes 2 coins
- 2.- Player 2 takes x coins ($x \geq 1$ and $x \leq 3$)
- 3.- Player 1 takes $4-x$ coins
- 4.- Repeat steps 2 and 3 until there is only 1 coin left.

Programming language: is a set of rules, symbols and special words used to build a program.

```
Void prueba() {  
    i:= 0;  
    while (i < 9) do {  
        printf("%d\n",i);  
        i++;  
    }  
    ...  
}
```



```
11100011000010000  
10011111010101010  
100001111010101  
01011100011010101  
1111111101110101  
01111101010101010  
101010  
...
```



Programming language: is a set of rules, symbols and special words used to build a program.

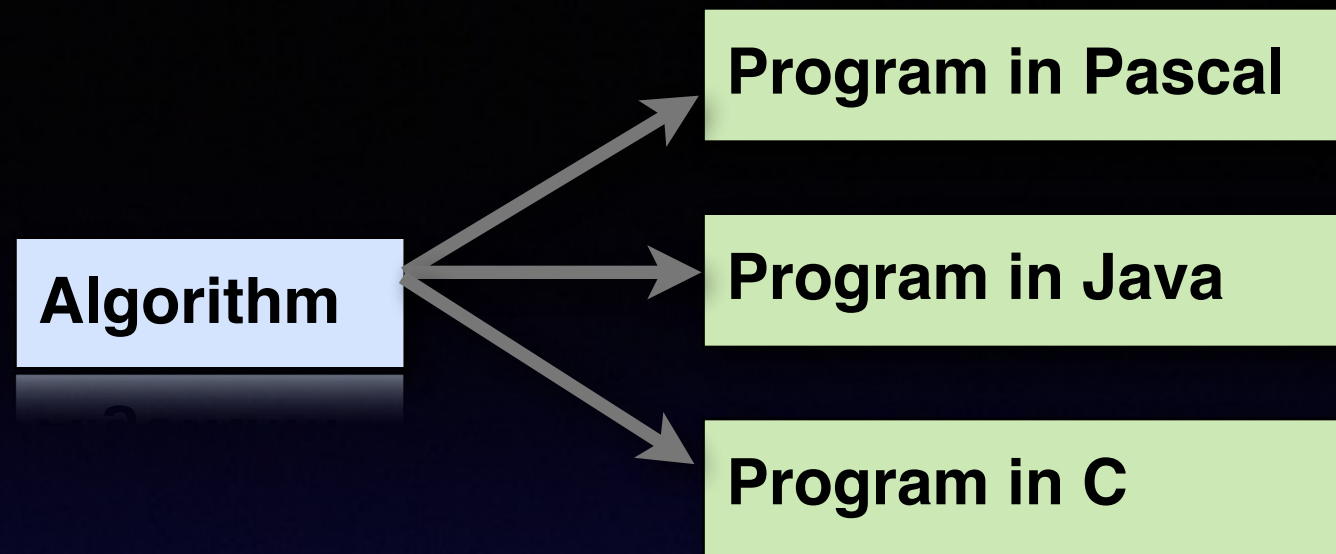
Program: Is a sequence of instructions that indicates the actions that have to be run by a computer.

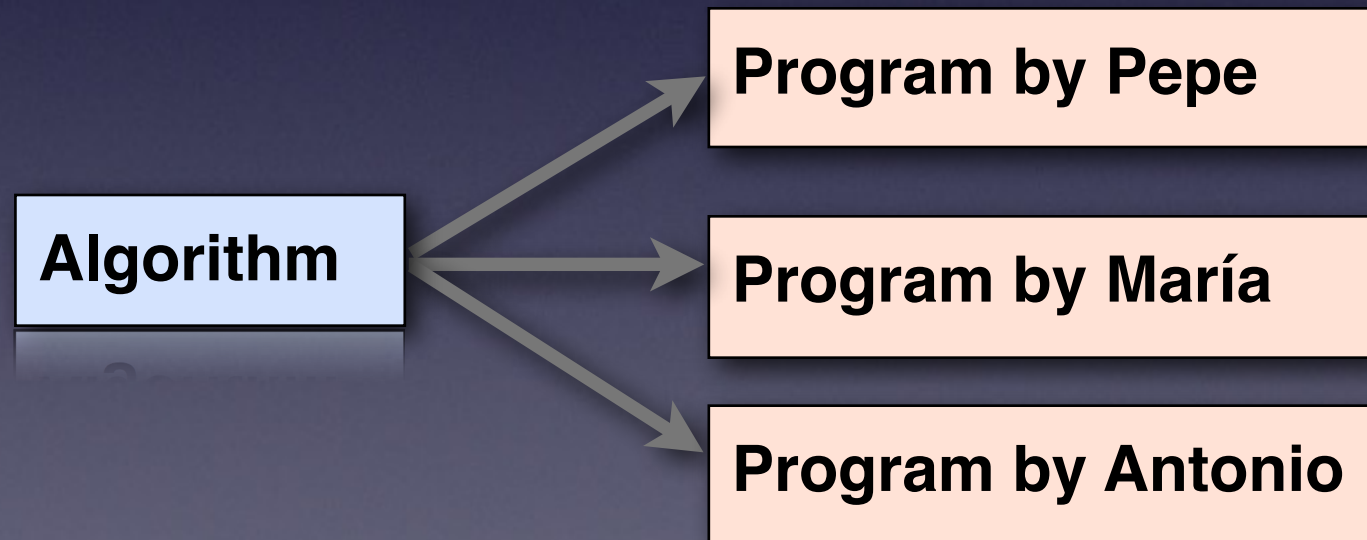
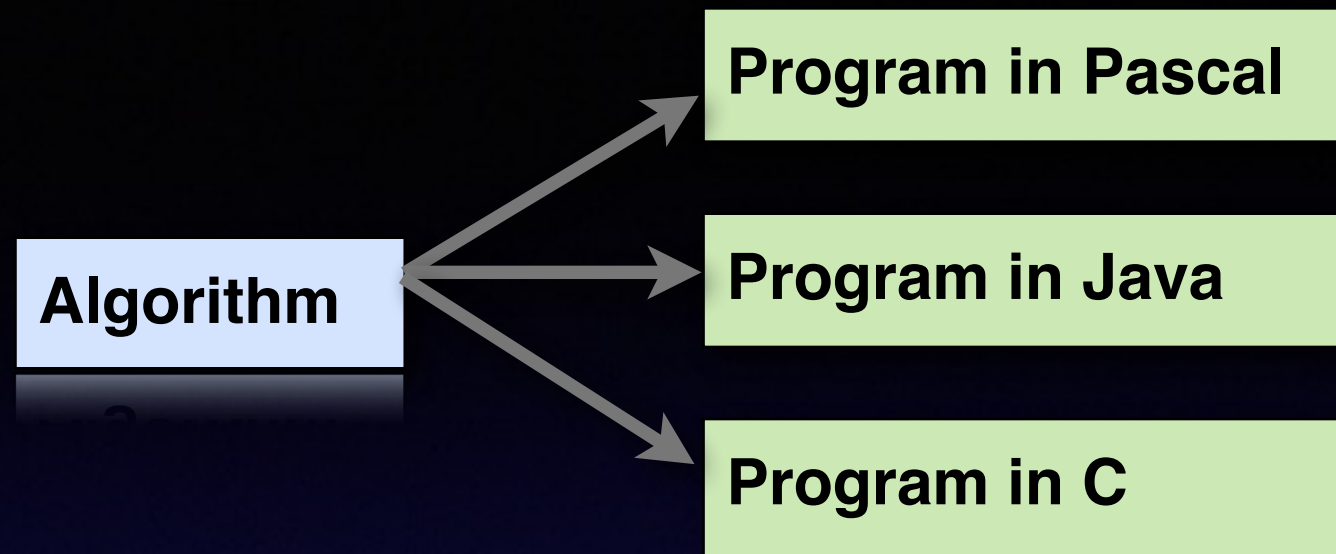
```
Void prueba() {  
    i:= 0;  
    while (i < 9) do {  
        printf("%d\n",i);  
        i++;  
    }  
    ...  
}
```

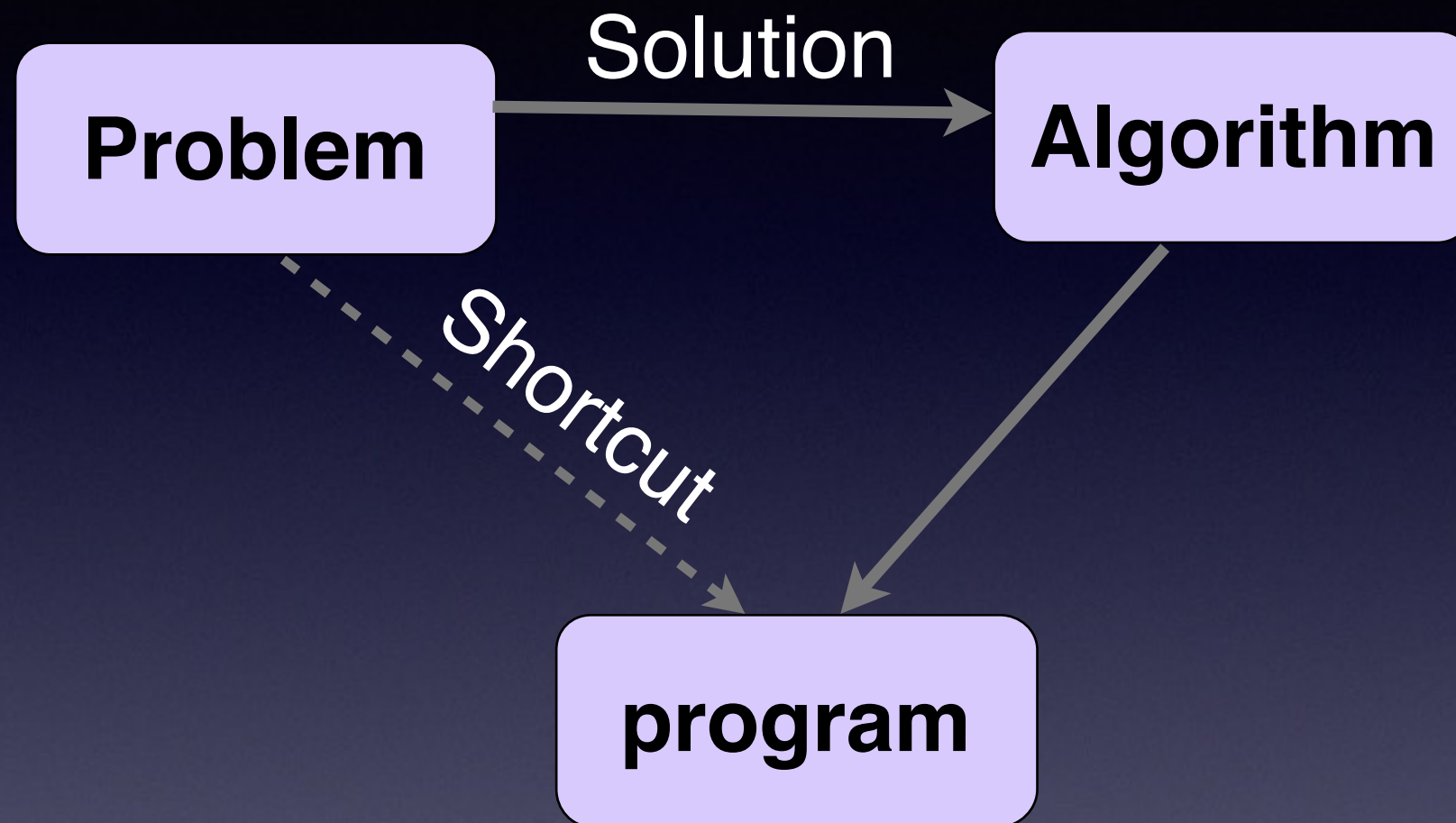


```
11100011000010000  
100111110101010  
100001111010101  
010111000110101  
1111111101110101  
011111010101010  
101010  
...
```









Application's life cycle:

An **application** is composed of one or several interrelated programs to carry out a certain task in an automatic way using a computer.



Application's life cycle:



Application life cycle: Is the process followed from the Identification of the problem or task until we have a solution installed and working on the users' computers, as long as the Application is useful for them.

Application's life cycle:



Design phases

Installation phases

Design phases:

Problem

Design phases:

Problem

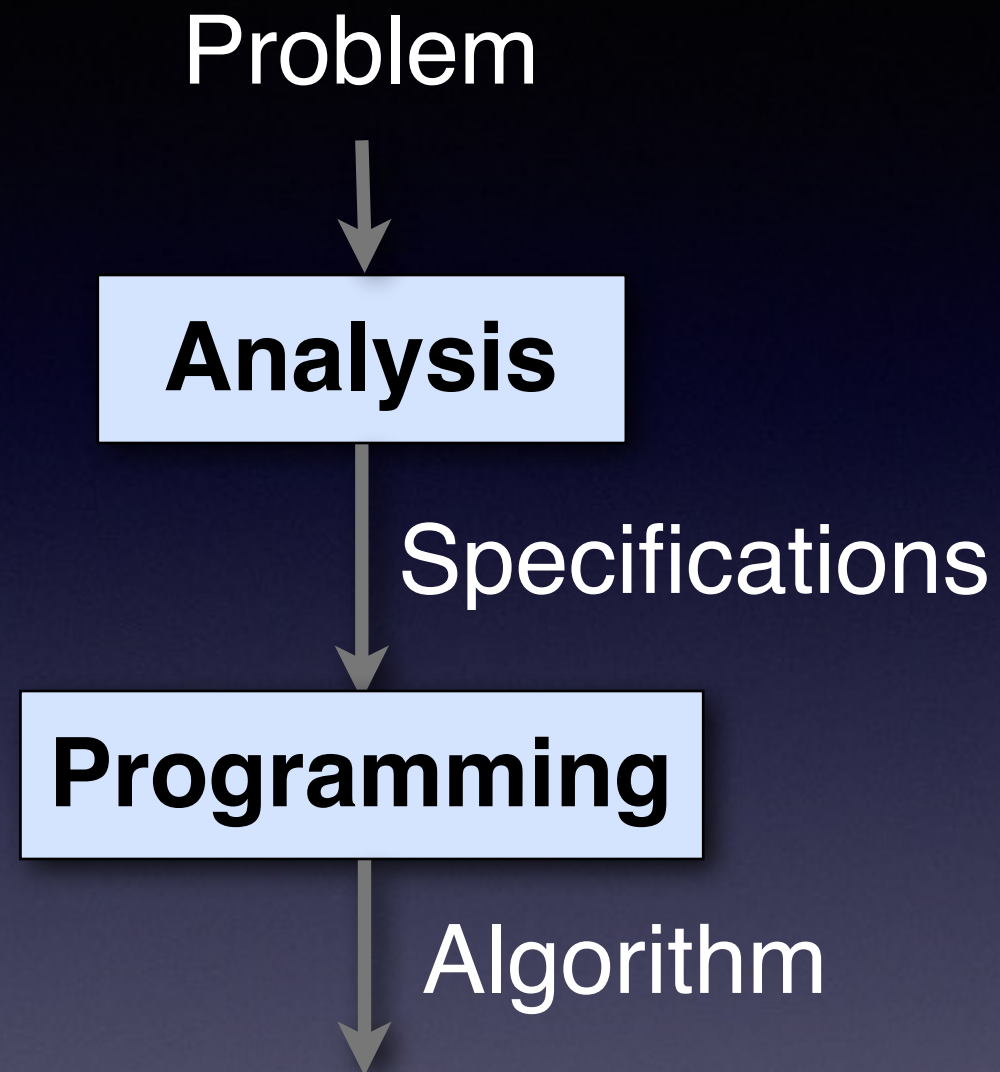


Analysis

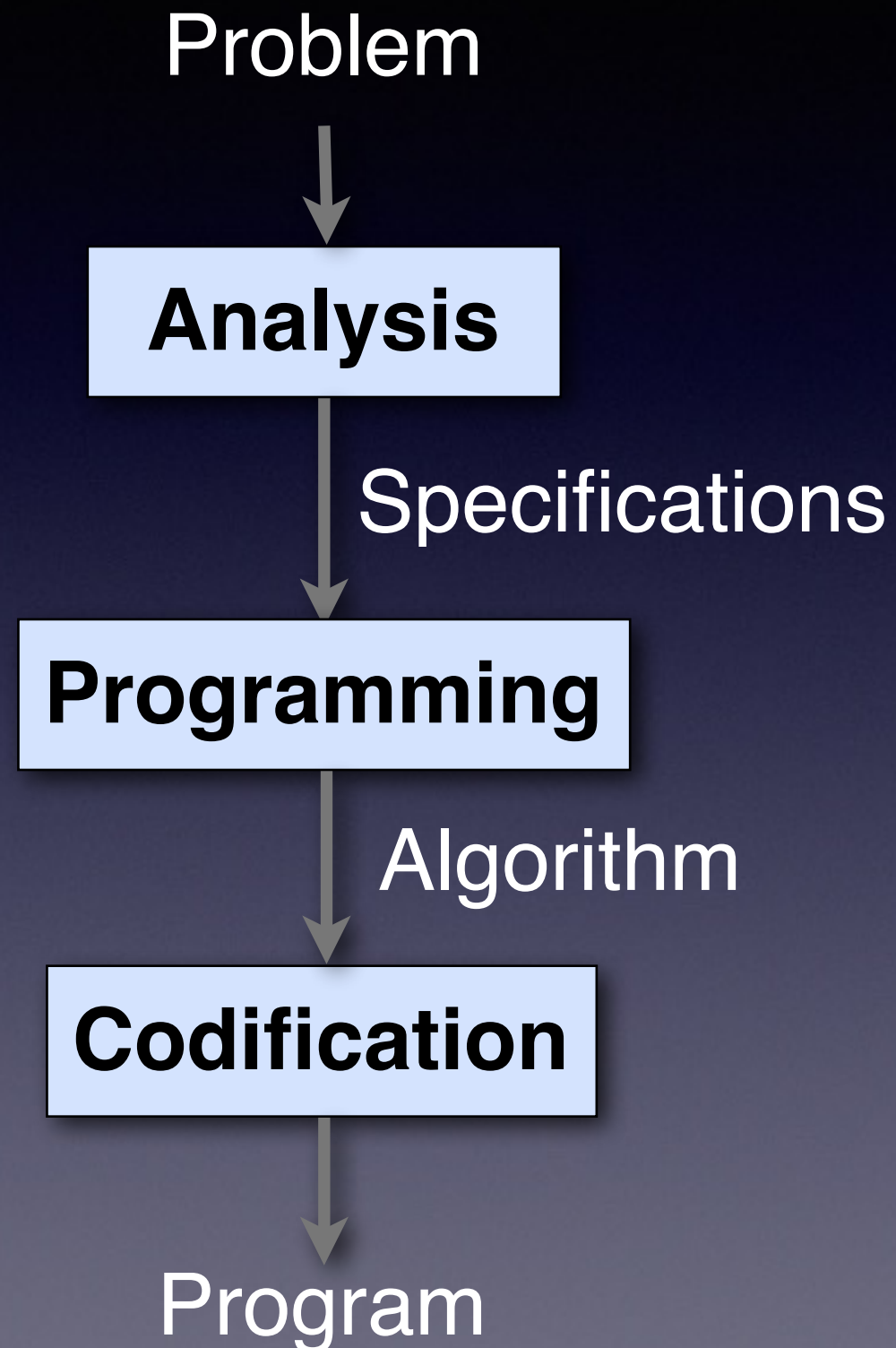


Specifications

Design phases:



Design phases:



Installation phases

Program

Design phases:

Problem



Analysis



Specifications

Programming

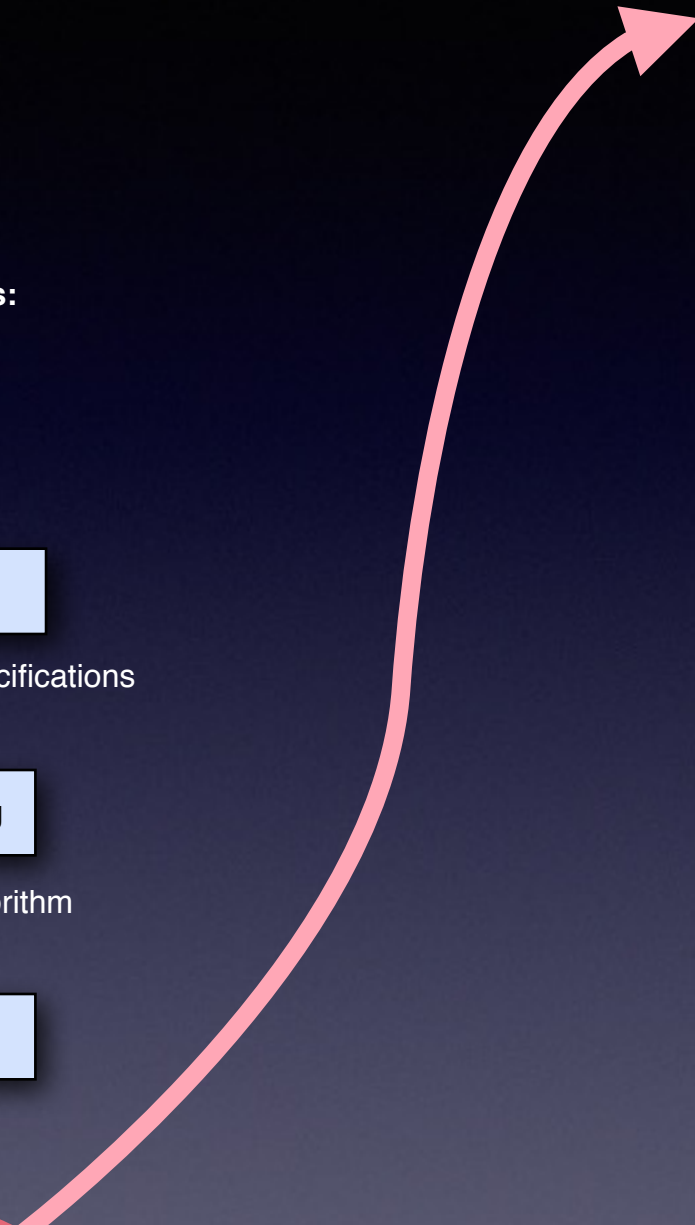


Algorithm

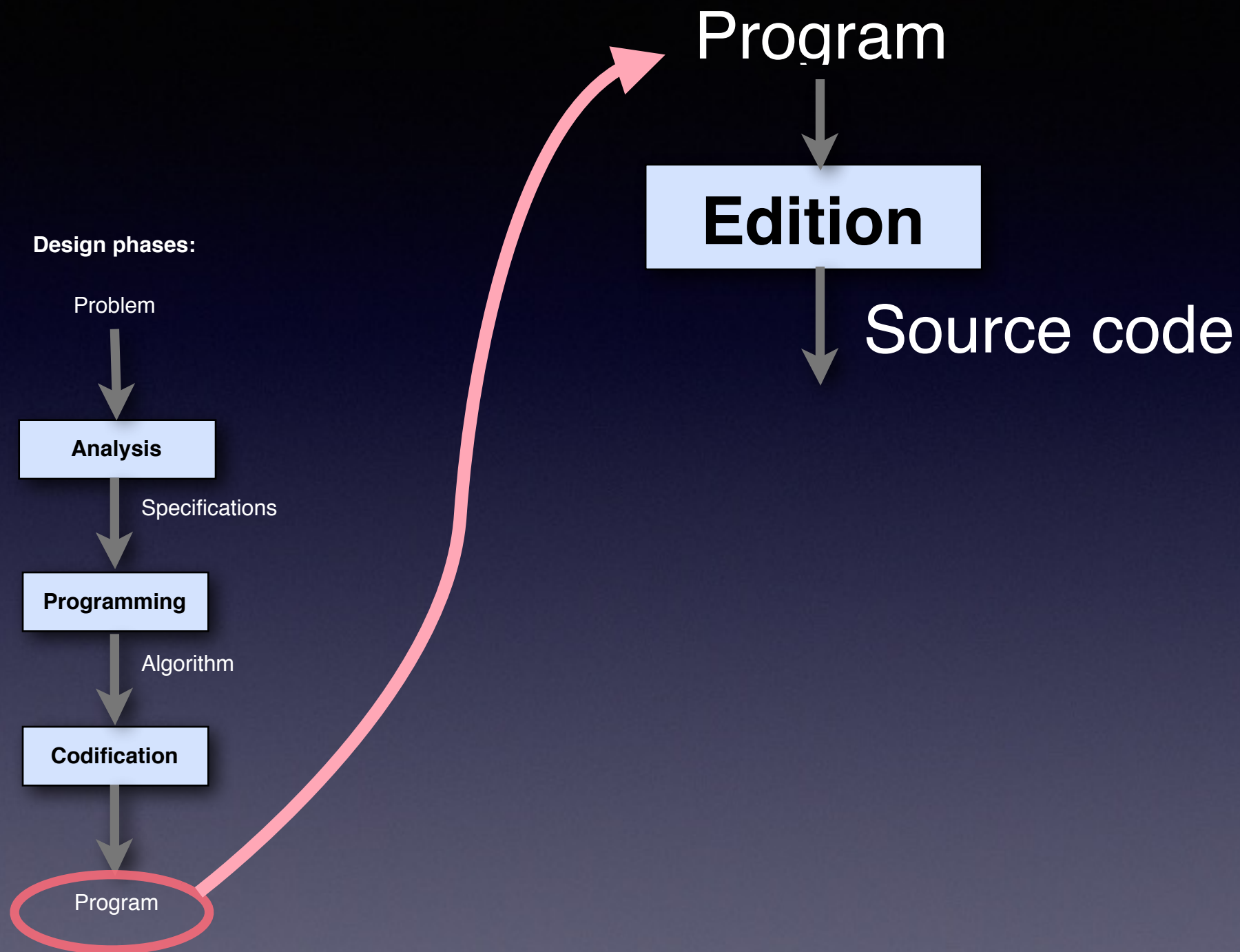
Codification



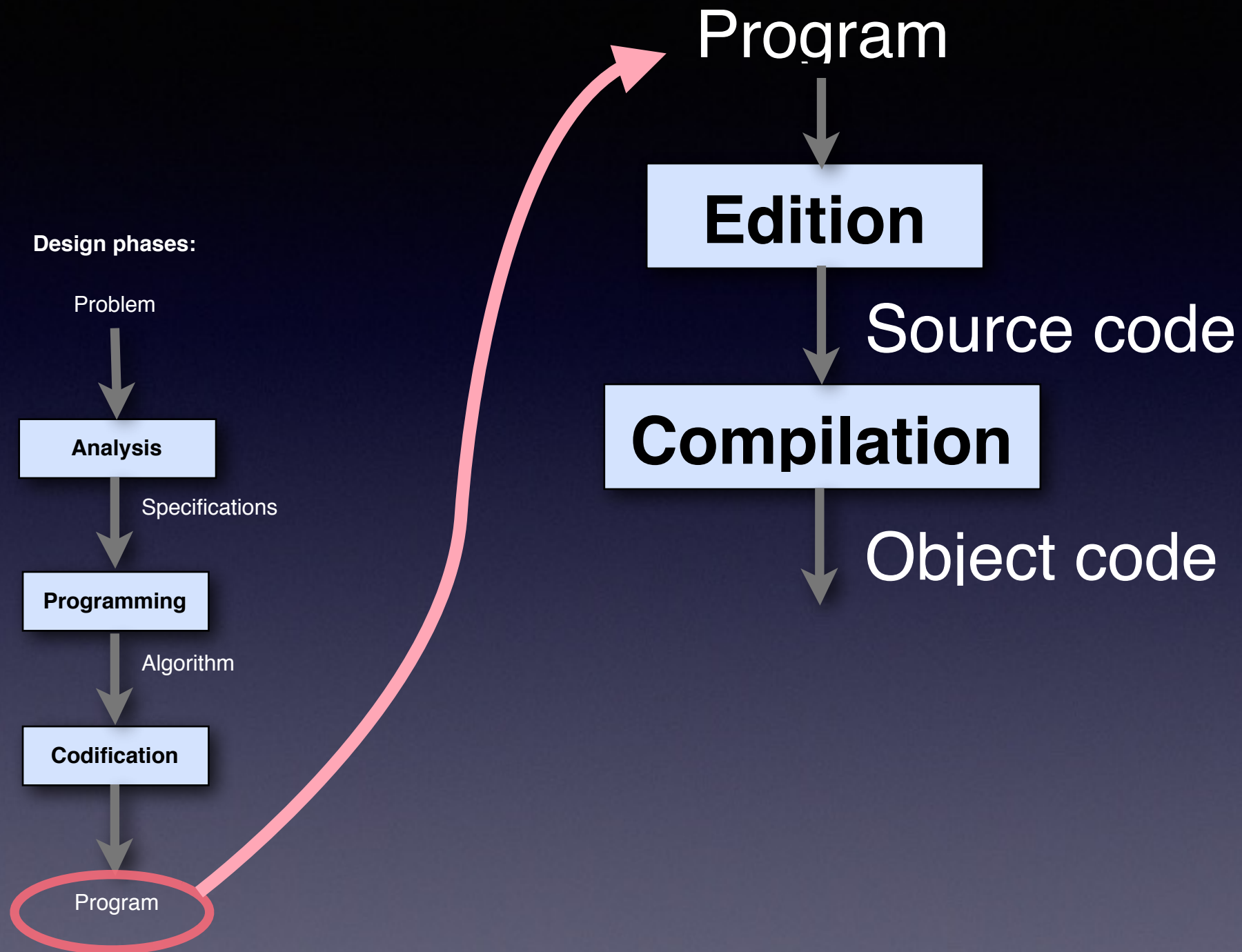
Program



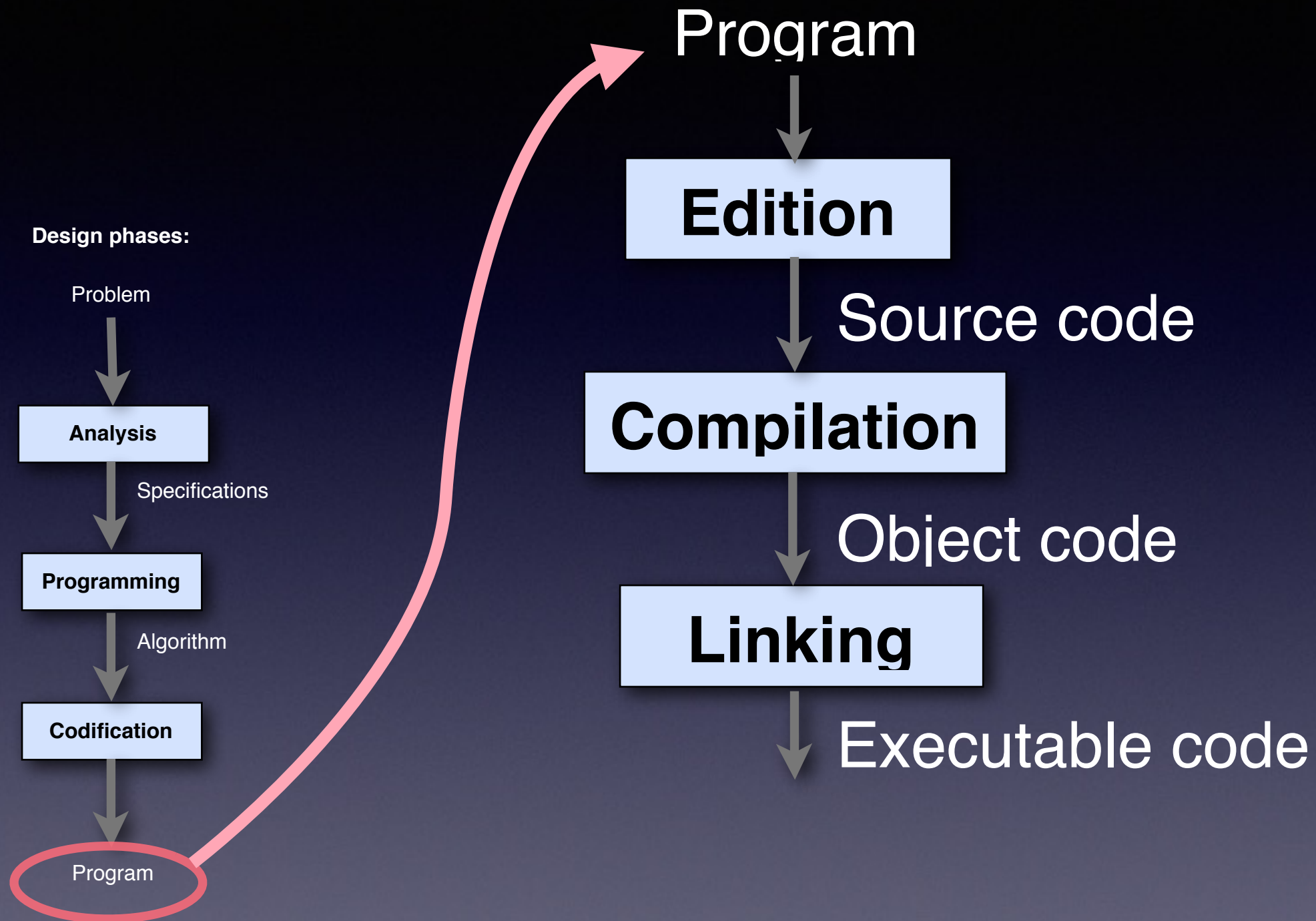
Installation phases



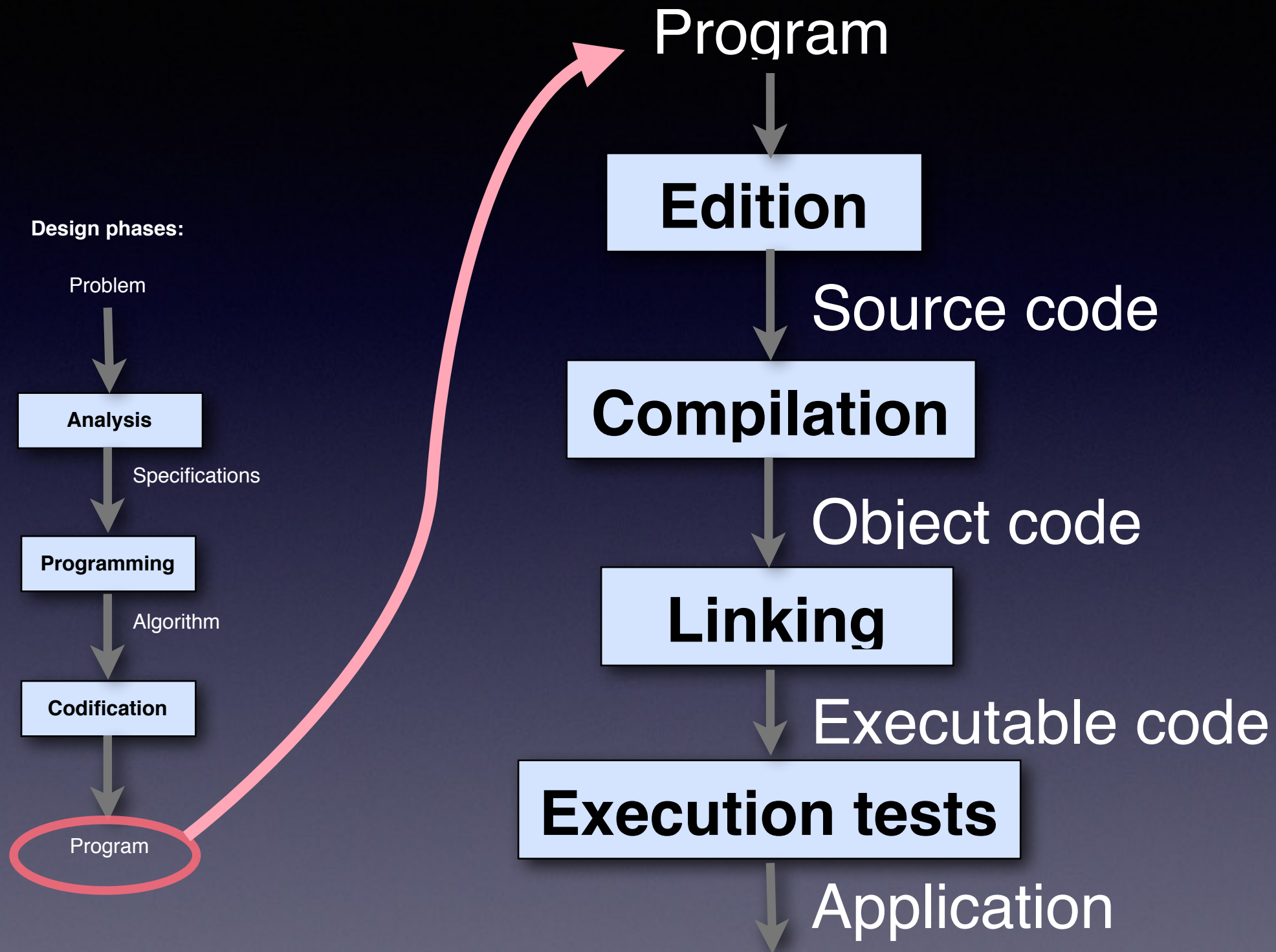
Installation phases



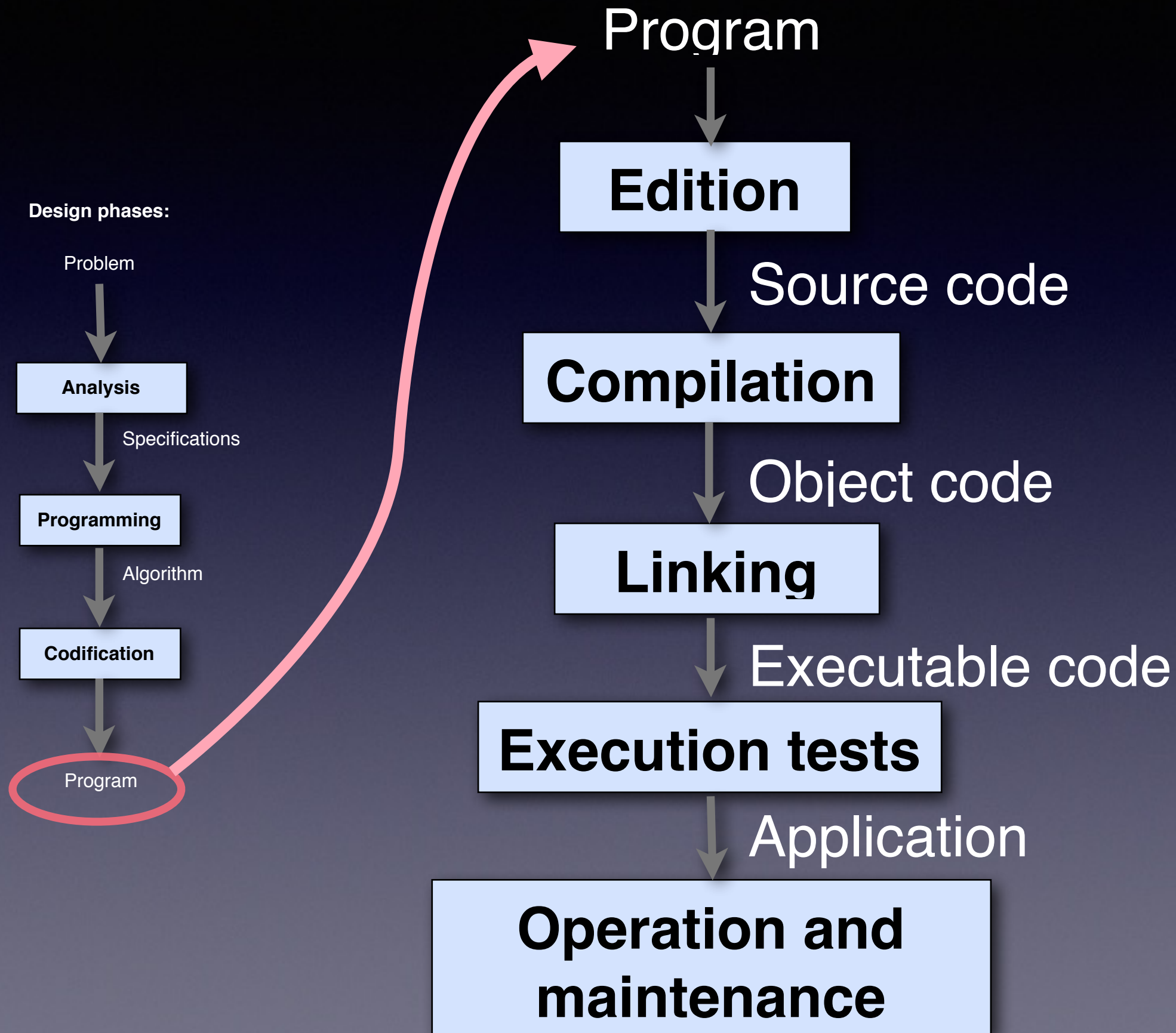
Installation phases



Installation phases

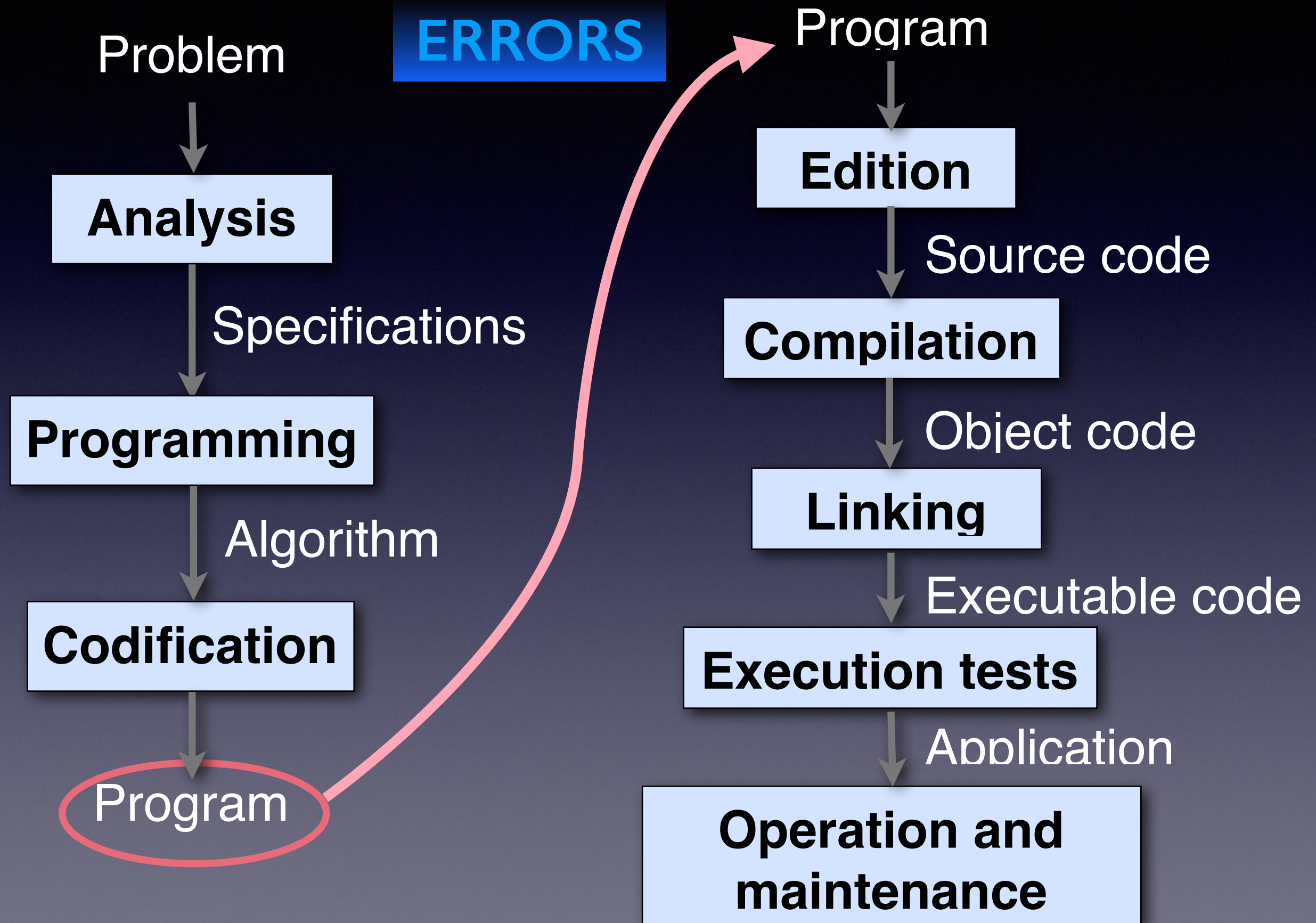


Installation phases



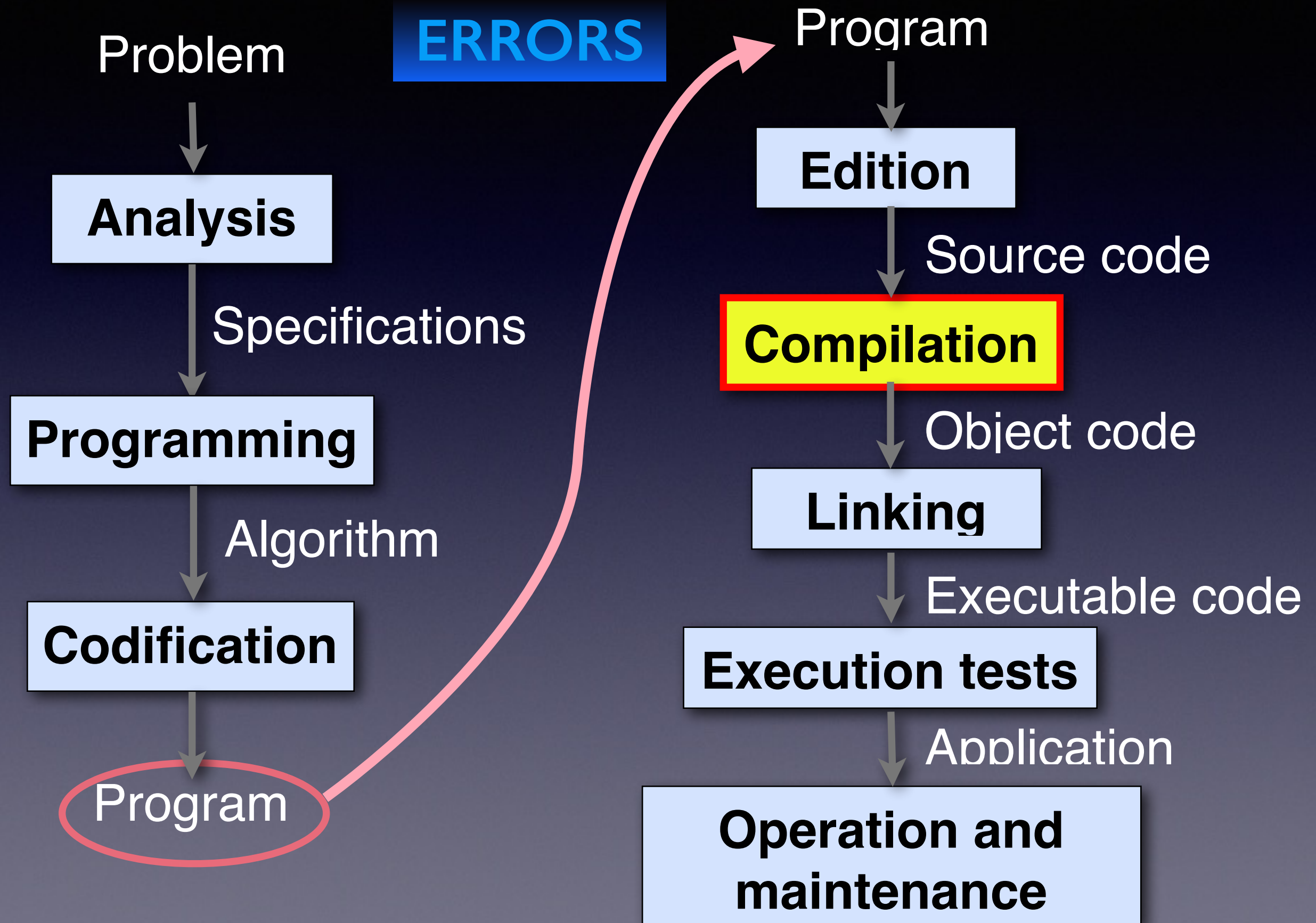
Design phases:

Installation phases



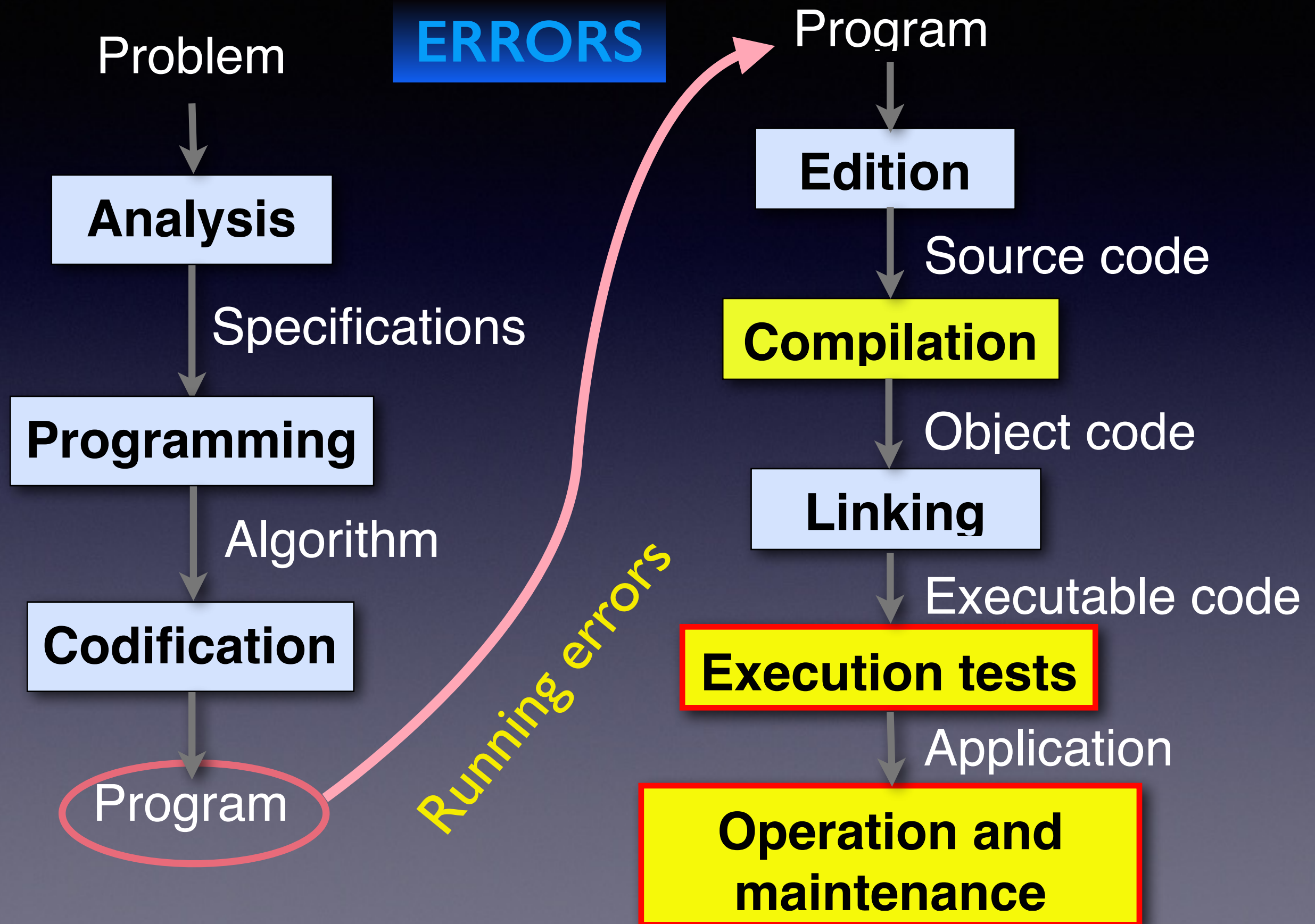
Design phases:

Installation phases



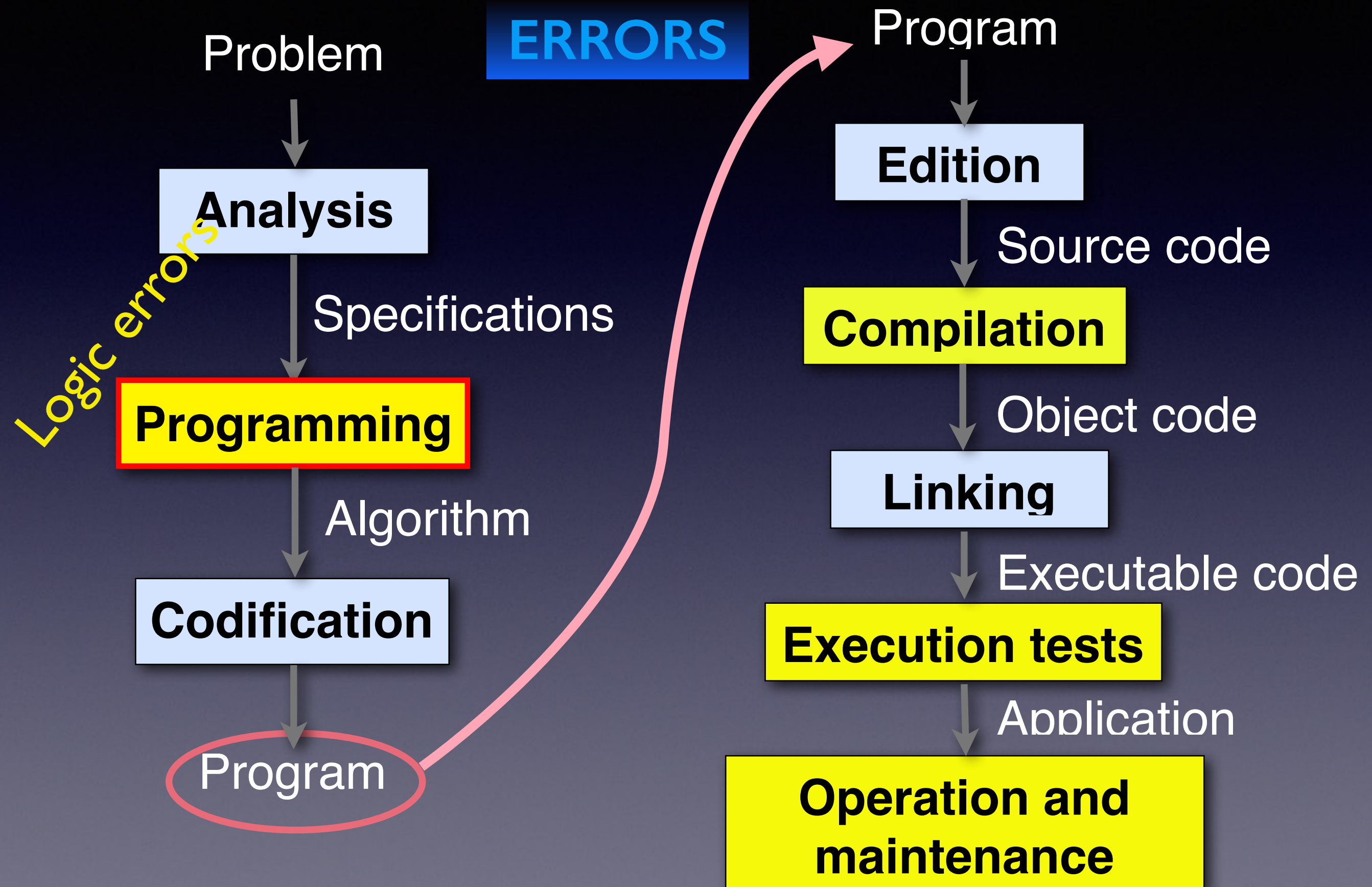
Design phases:

Installation phases



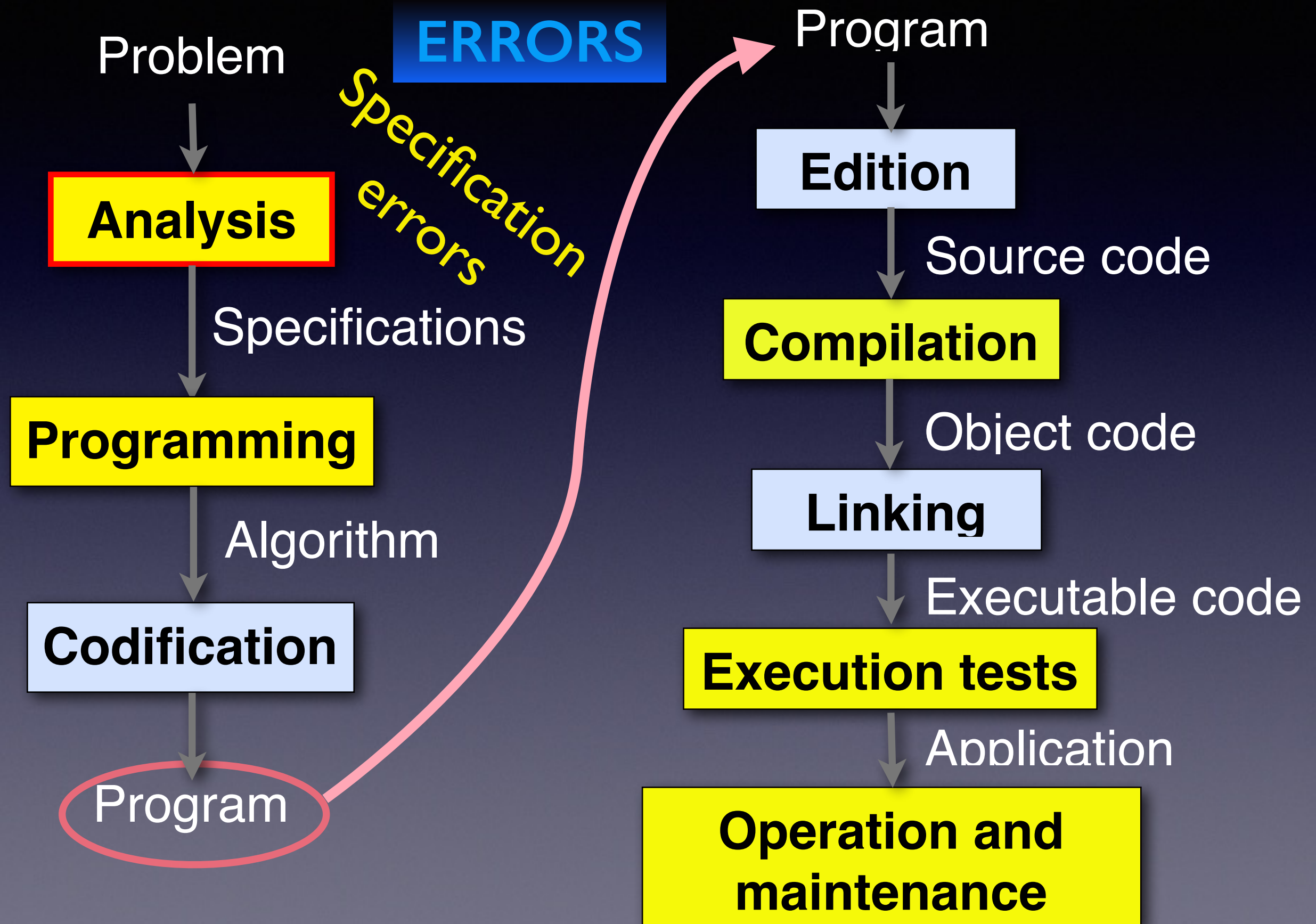
Design phases:

Installation phases



Design phases:

Installation phases



Quality of the programs

Quality of the programs



Legibility

Quality of the programs



Legibility



Portability

Quality of the programs



Legibility



Portability



(Easy to write in different languages)

Quality of the programs



Legibility



Portability



Modifiability

Quality of the programs



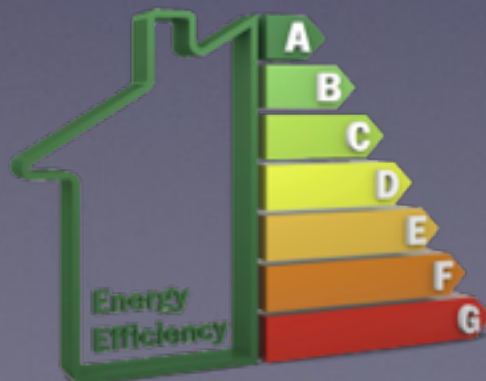
Legibility



Portability



Modifiability



Efficiency

Quality of the programs



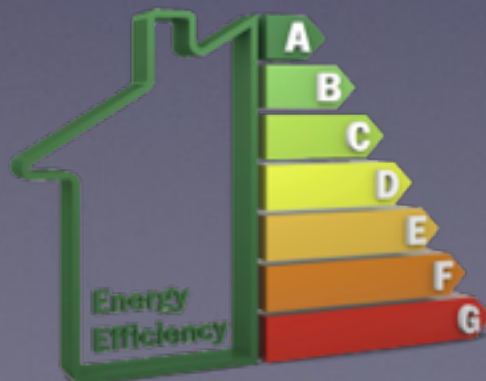
Legibility



Portability



Modifiability



Efficiency

Programming methodology is the set of methods and techniques that help to develop programs that fulfill all the quality requirements exposed above.



Documentation of Programs

Internal documentation

```
Void prueba() {  
    i:= 0;  
    while (i < 9) do {  
        printf("%d\n",i);  
        i++;  
    }  
    ...  
}
```



Documentation of Programs

Internal documentation

```
Void prueba() {  
    i:= 0;  
    while (i < 9) do {  
        printf("%d\n",i);  
        i++;  
    }  
    ...  
}
```

Comments:

```
// This is comment  
/* This is a comment with  
two or more lines */
```

Autocommented code:

Adequate and meaningful identifiers

Declaration of constants for fixed values

Indentation, empty lines, etc.



Documentation of Programs

Internal documentation

```
Void prueba() {  
    i:= 0;  
    while (i < 9) do {  
        printf("%d\n",i);  
        i++;  
    }  
    ...  
}
```

External documentation



- Specifications from the analysis.
- Description and diagrams of the design of the programs
- User's manual
- Maintenance manual