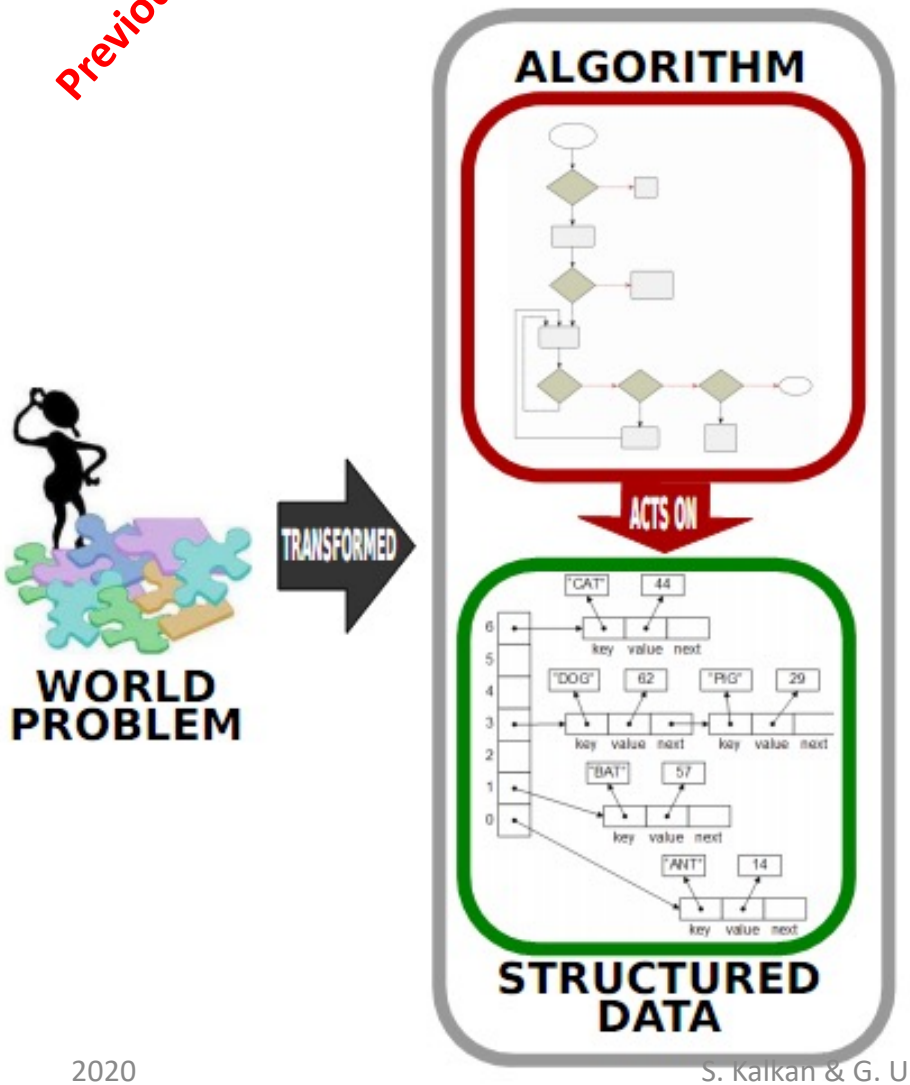
Ceng 111 – Fall 2021
Week 6a

Credit: Some slides are from the “Invitation to Computer Science” book by G. M. Schneider, J. L. Gersting and some from the “Digital Design” book by M. M. Mano and M. D. Ciletti.

Previously on CENG111!

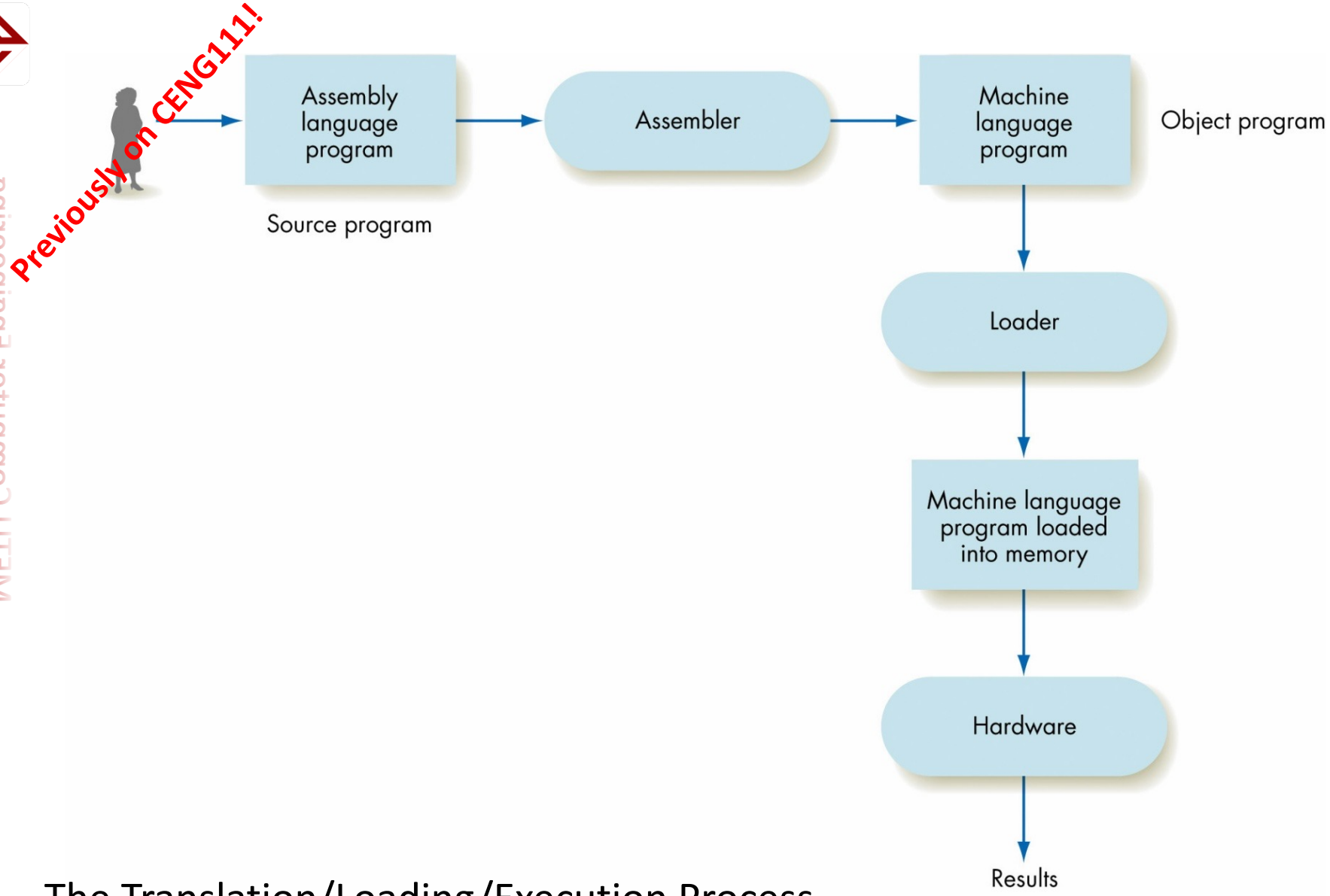
Program, Programming



IMPLEMENTED

```
int alice = 1
int bob = 456,
int carol;
main(void)
{
 carol = alice*bob;
 printf("%d", carol);
}
```

PROGRAM



The Translation/Loading/Execution Process

```
01010101 01001000 10001001 11100101 10001011 00010101 10110010 00000011
00100000 00000000 10001011 00000101 10110000 00000011 00100000 00000000
00001111 10001111 11000010 10001001 00000101 10111011 00000011 00100000
00000000 00111000 00000000 00000000 00000000 00000000 11001001 11000011
...
11001000 00000001 00000000 00000000  00000000 00000000
```

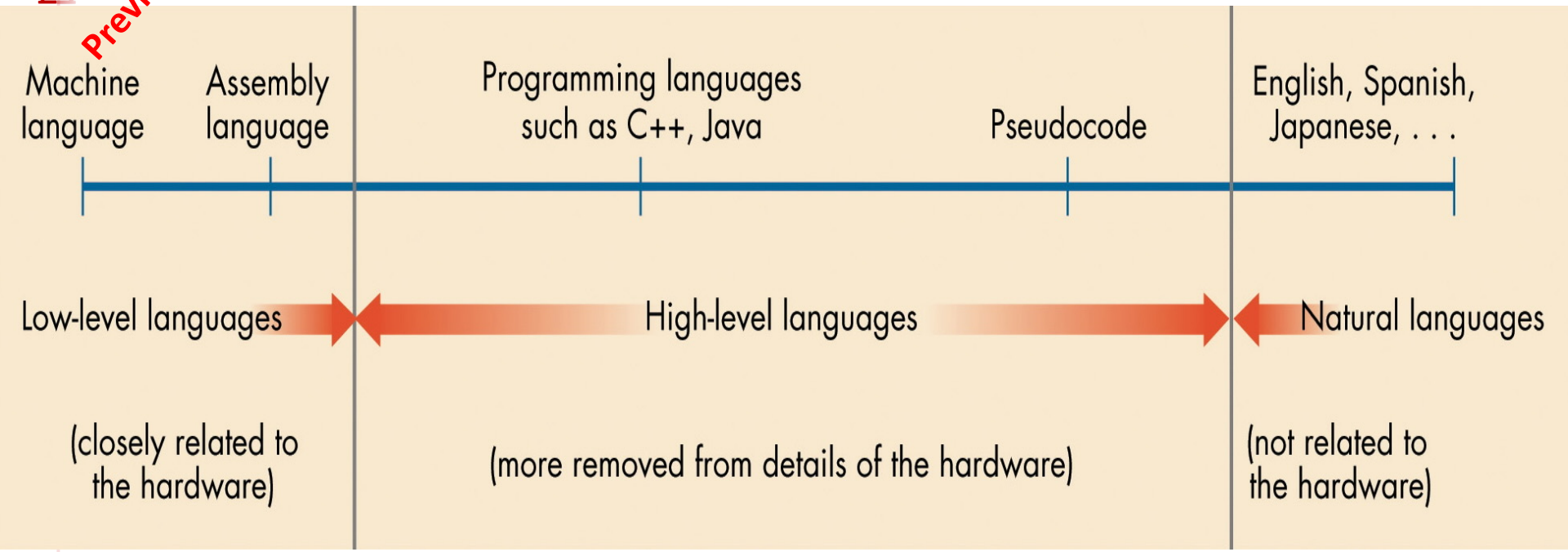
Previously on CENG111!

```
main:
        pushq   %rbp
        movq    %rsp, %rbp
        movl    alice(%rip), %edx
        movl    bob(%rip), %eax
        imull   %edx, %eax
        movl    %eax, carol(%rip)
        movl    $0, %eax
        leave
        ret
alice:
        .long   123
bob:
        .long   456
```

```
int alice = 123;
int bob = 456;
int carol;
main(void)
{
  carol = alice*bob;
}
```

The Spectrum of Programming Languages

Previously on CENG111!



- There is a limit to how high a language can get.
- Why can't we write programs in our spoken language?

Previously on CENG111!

Programming Language Paradigms

- Classification / Categorization of programming languages.
  - Imperative Paradigm
  - Functional Paradigm
  - Logical-declarative Paradigm
  - Object-oriented Paradigm
  - Concurrent Paradigm
  - Event-driven Paradigm

Imperative Paradigm

Statement_1

Statement_2

Statement_3

Statement_4

Statement_5

From C:

```
int a = 2;
int b = a * 2;
int c;

c = -b – sqrt(b*b - 4*a*c) / (2*a);
```

Previously on CENG111!

METU Computer Engineering

Functional Paradigm

- Data environment is restricted.
- Functions receive their inputs and return their results to the data environment.
- Programmer's task:
  - decompose the problem into a set of functions such that the composition of these functions produce the desired result.

Previously on CENG111!

Logical-declarative Paradigm

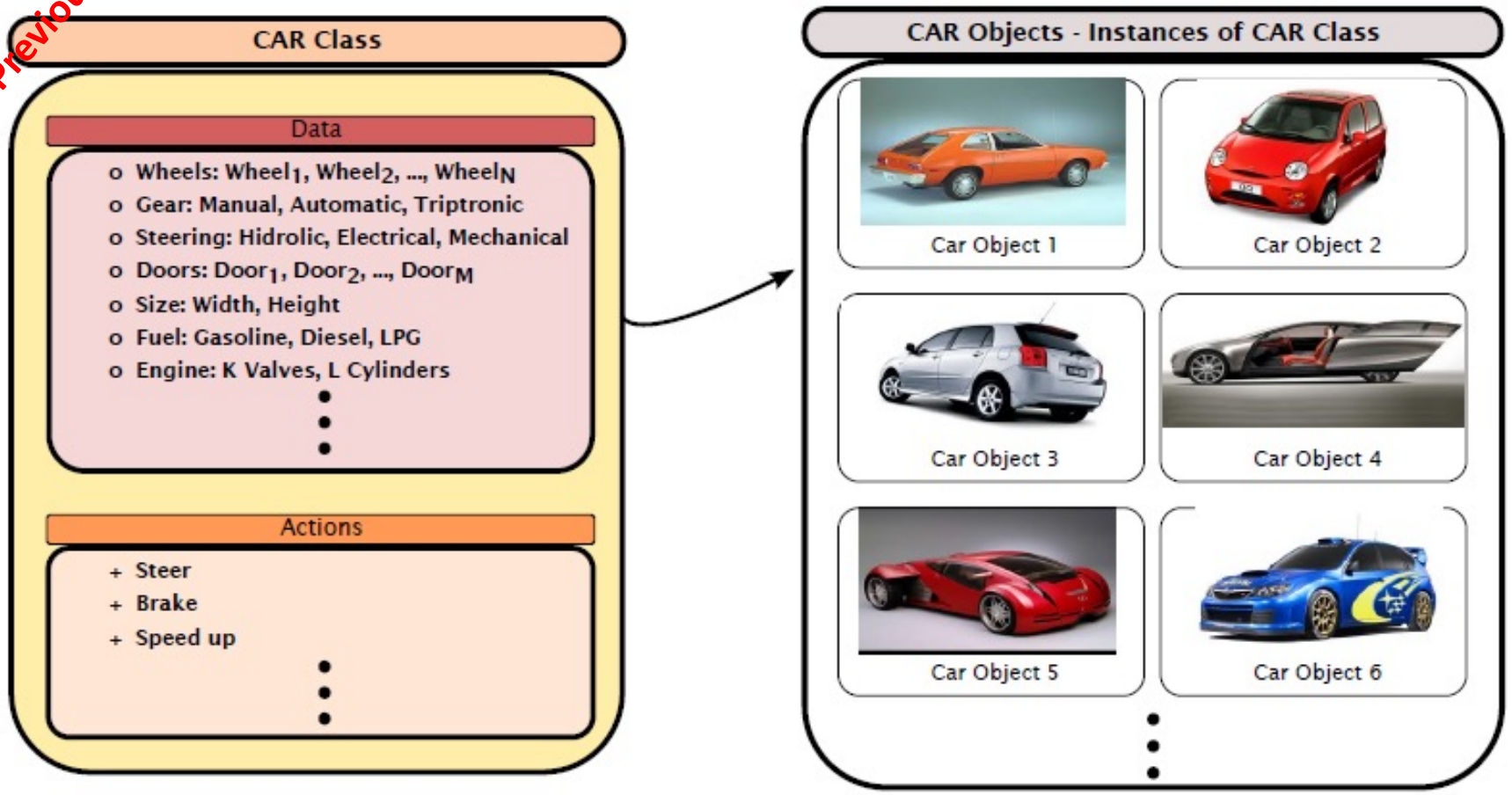
- The data and the relations are states as rules, or facts.
- The problem is solved by writing new rules/facts.

From Prolog:

```
mother(matilda,ruth).
mother(trudi,paggy).
mother(eve,alice).
mother(zoe,sue).
mother(eve,trudi).
mother(matilda,eve).
mother(eve,carol).
grandma(X,Y) :- mother(X,Z), mother(Z,Y).
```

Previously on CENG111!

Object Oriented Paradigm



Concurrent Paradigm

- Programming using multiple CPUs concurrently.
- The task is to assign the overall flow & data to individual CPUs.
- With the bottleneck in CPU power, this paradigm is going to be the trend in the future.

Event-Driven Paradigm

- A program is composed of events and what to do in case of events.
- The task is to decompose a problem into a set of events and the corresponding functionalities that will be executed in case of events.
- Suitable for Graphical User Interface design.

Previously on CENG111!

The hyperspace of languages

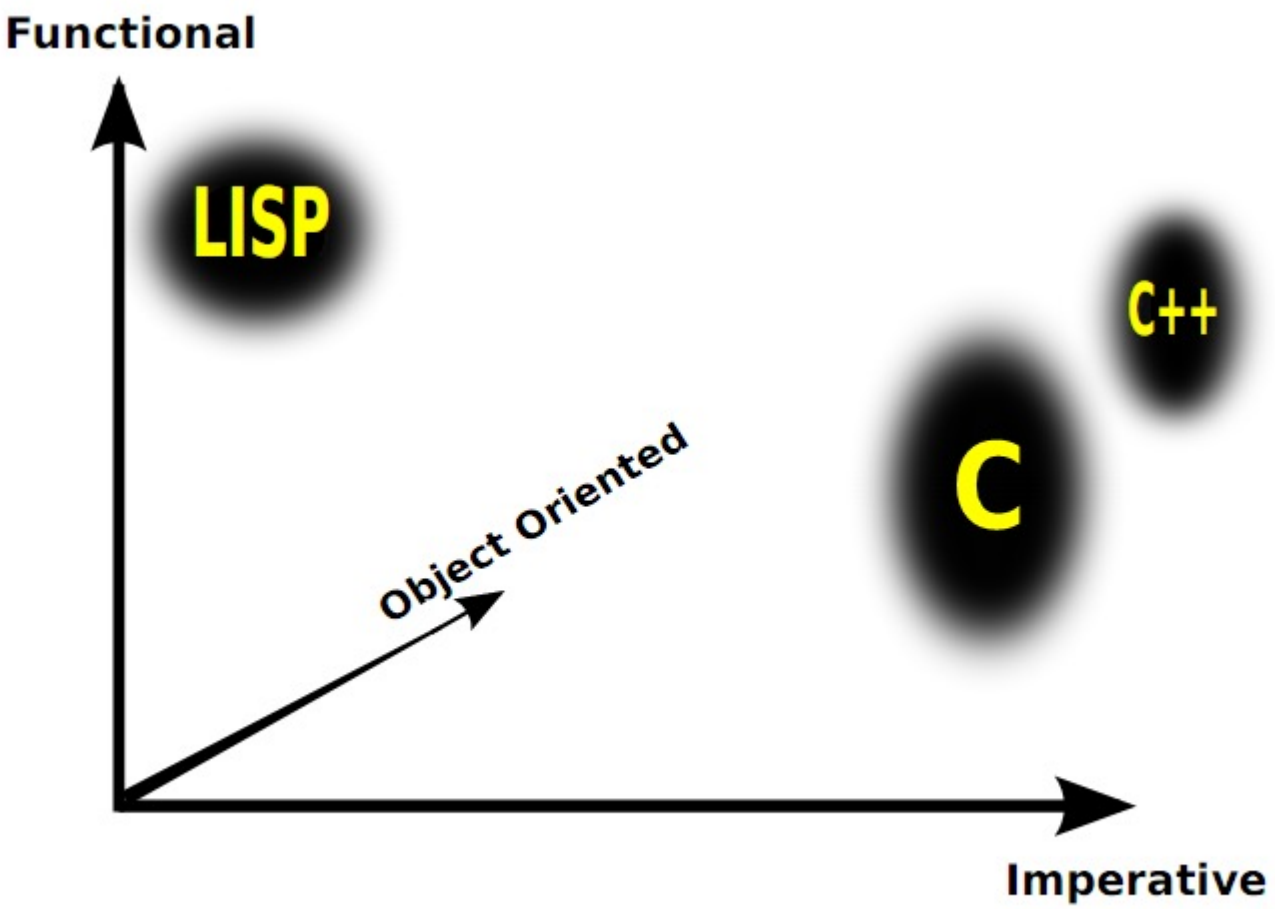


Figure 1.4: The hyperspace of programming *(only 3 axes displayed)*

Previously on CENG111!

Choosing a PL

- Ex: Moving soil with a shovel and a grader

Factors that affect choosing a PL

- Domain & Technical Nature of the Problem

a) Finding the pixels of an image with RGB value of [202,130,180] with a tolerance of 5.4% in intensity.

b) A proof system for planar geometry problems.

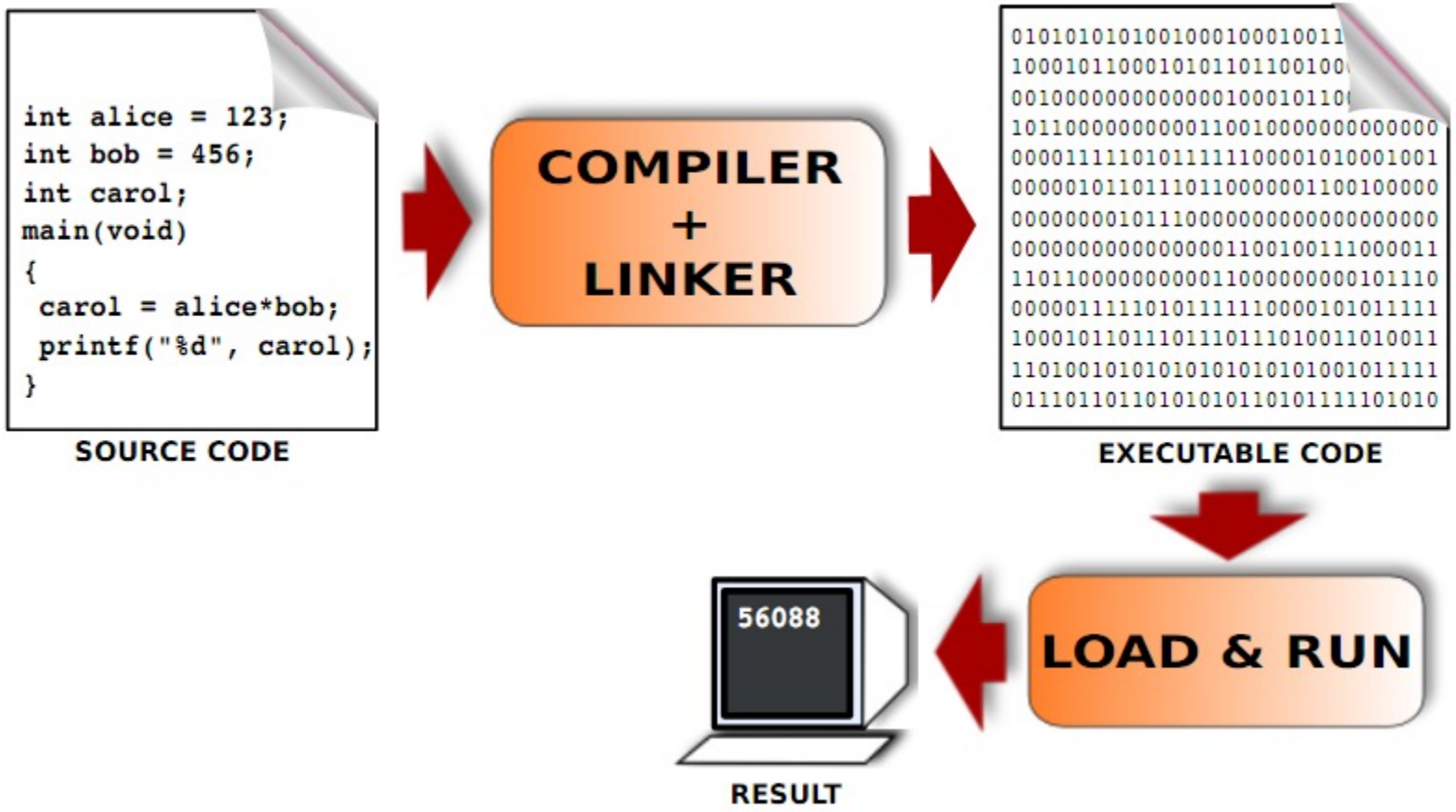
c) A computer game platform which will be used as a whole or in parts and may get extended even rewritten by programmers at various levels.

d) Payroll printing.

Factors that affect choosing a PL

- Personal taste and preference
- Circumstance-imposed constraints
  - e.g., time limit.
- Current trend

How are languages implemented

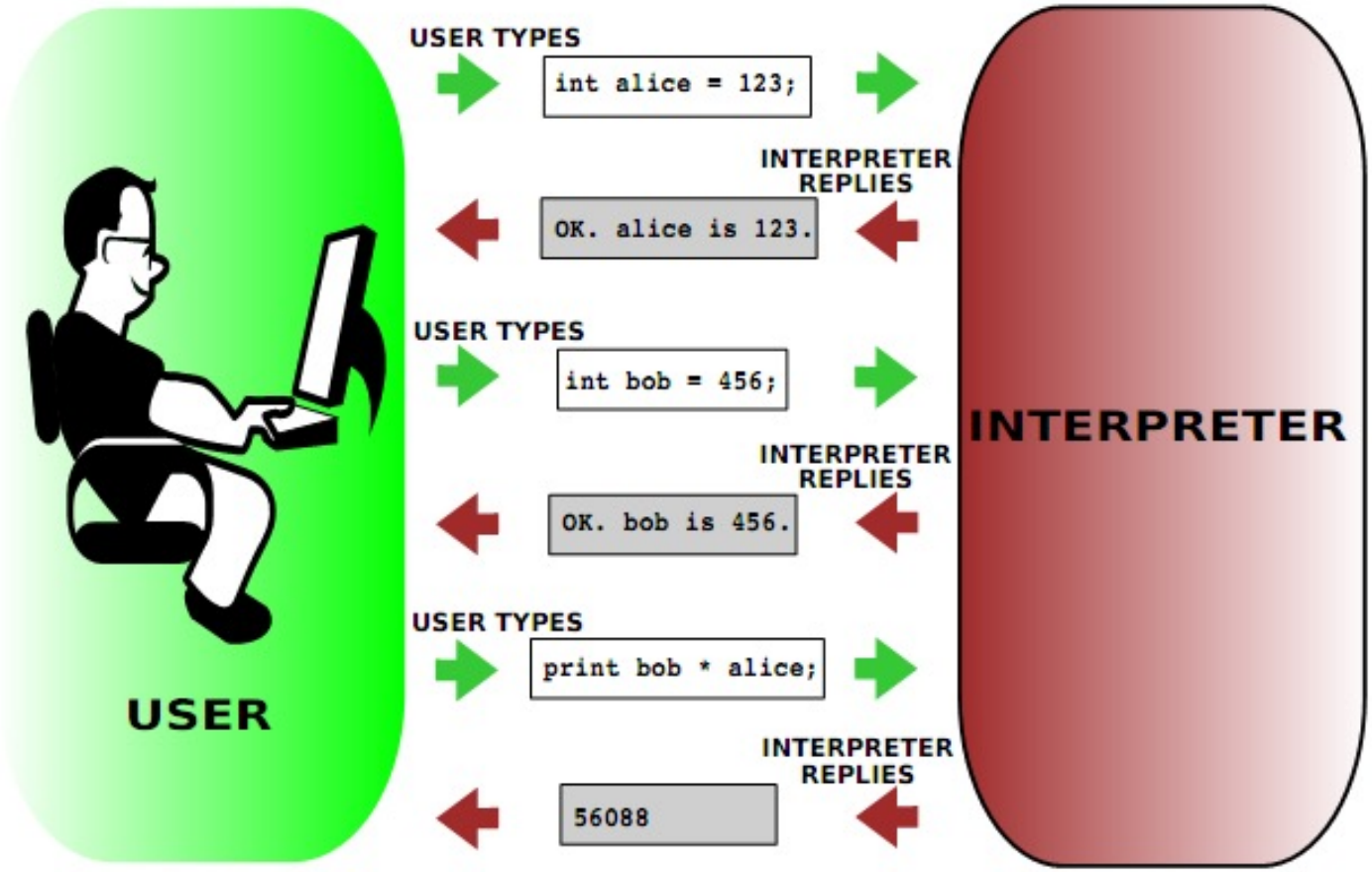


How are languages implemented

Previously on CENG111!

METU Computer Engineering

INTERPRETIVE APPROACH



Today

- The world of programming

Administrative Notes

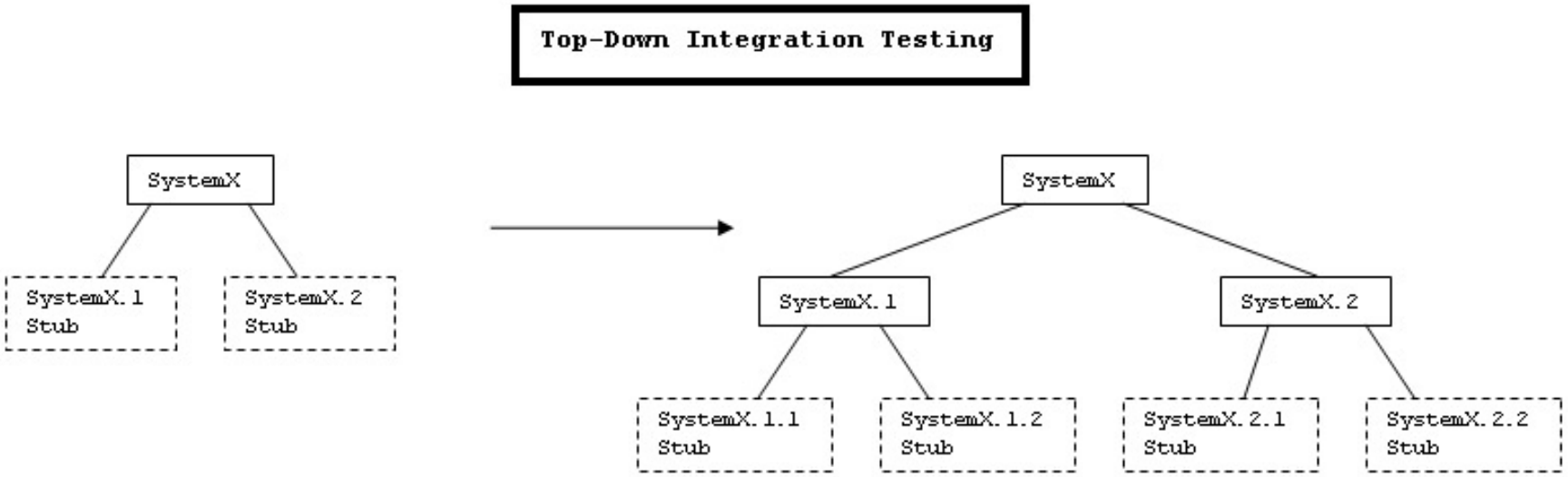
- Scratch assignment
- Midterm date:
  - 22 December, Wednesday, 18:00

How is a program written?

- Modular & Functional Breakdown
- For example:
  - User interface module
  - Database module
  - Control module

Testing

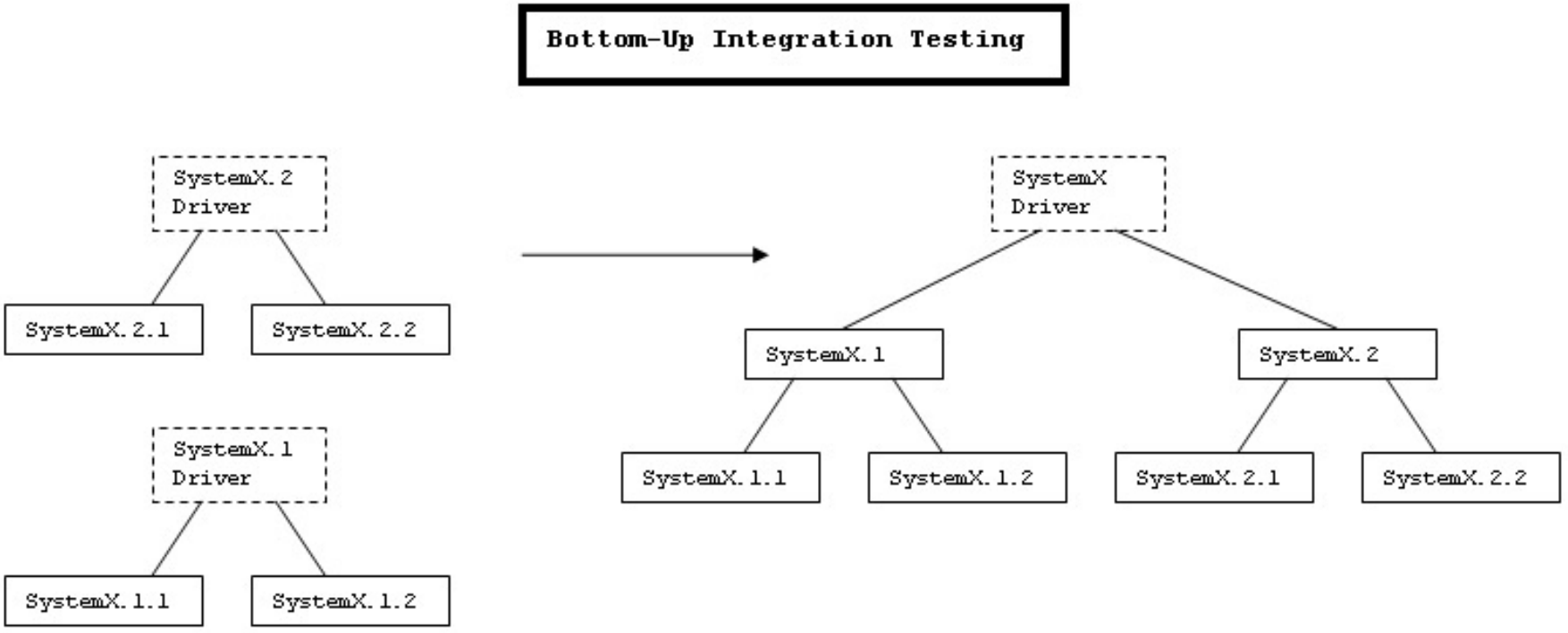
- Top-down Testing



http://sce.uhcl.edu/whiteta/sdp/subSystemIntegrationTesting.html

Testing

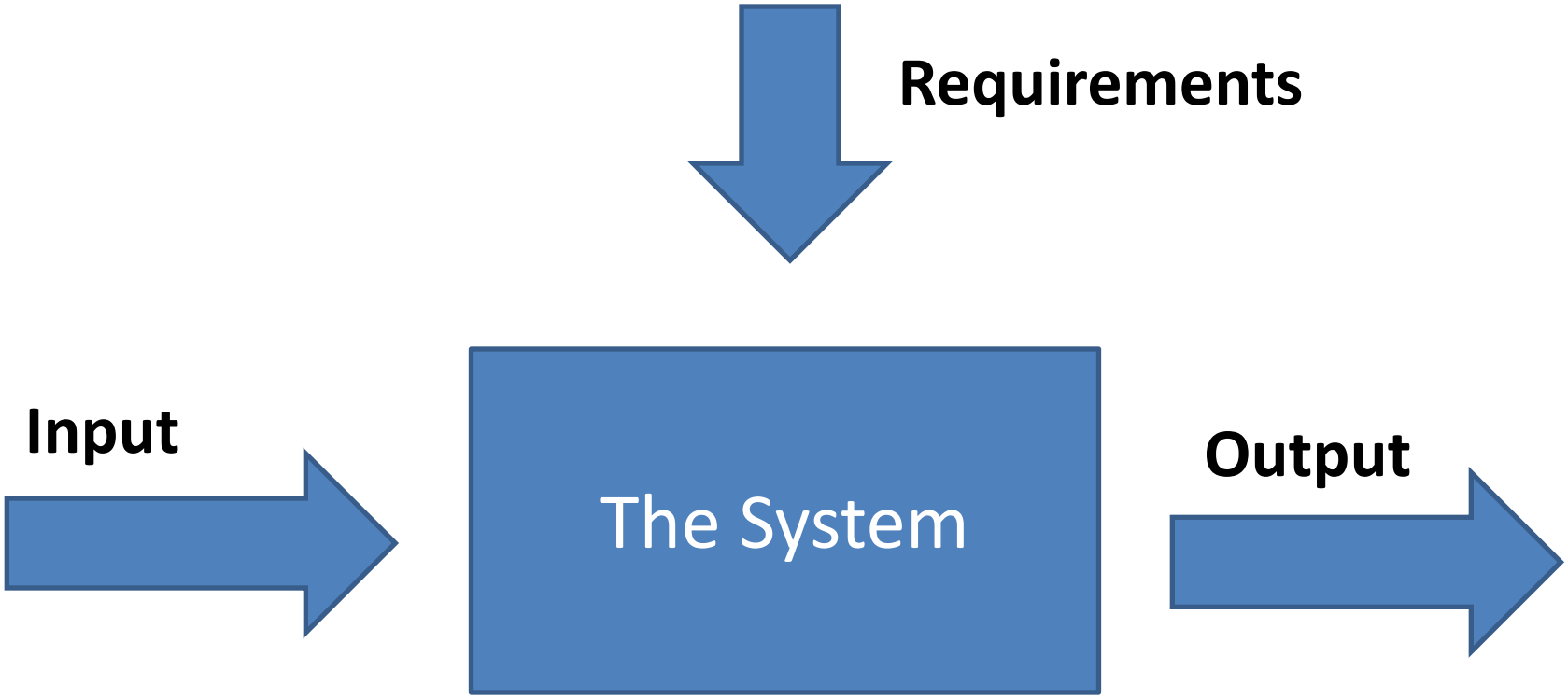
- Bottom-up Testing



http://sce.uhcl.edu/whiteta/sdp/subSystemIntegrationTesting.html

Testing

- Black-box Testing

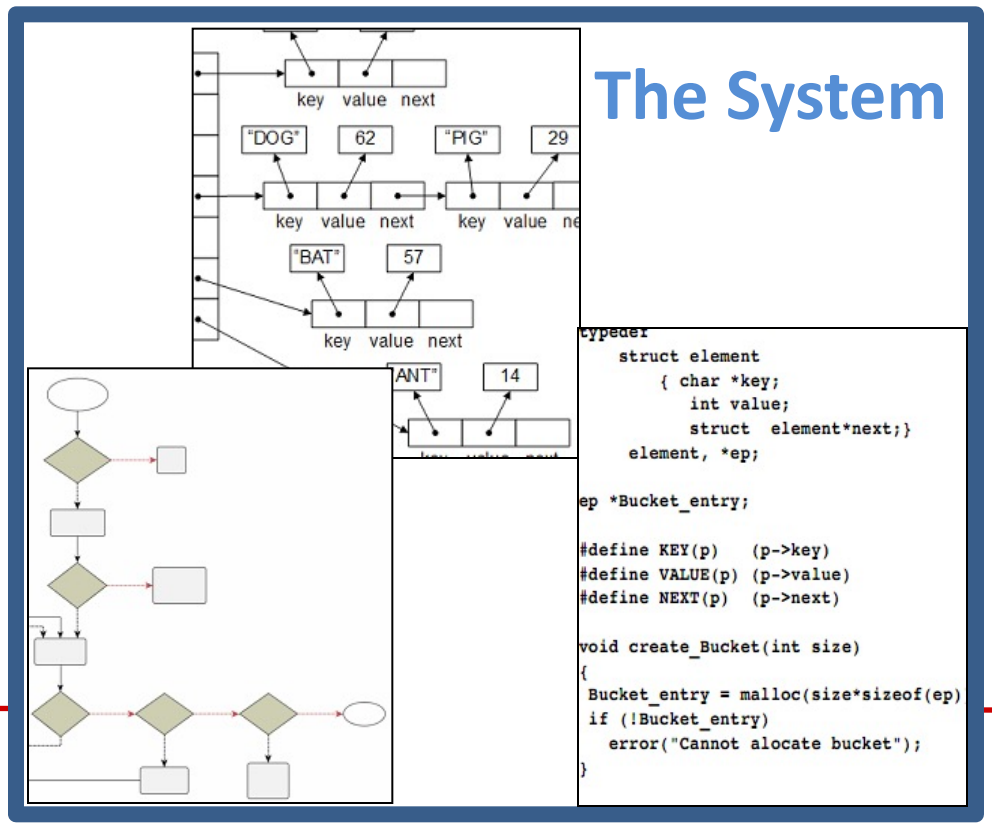


Testing

- White-box Testing

Requirements

Input



Output

Bugs, Errors

- Syntax Errors

  Area = 3.1415 * R * R

  Area = 3.1415 x R x R

- Run-time Errors

```
>>> def SqrtDelta(a,b,c):
>>>       return sqrt(b*b - 4*a*c)
>>>
>>> print SqrtDelta(1,3,1)
2.236067977497898
>>> print SqrtDelta(1,1,1)
ValueError: math domain error
```

Bugs, Errors

- Logical Errors

$$root_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a}$$

```
>>> root1 = (- b + sqrt(b*b - 4*a*c)) / 2*a
```

- Design Errors

$$x^3 + ax^2 + bx + c = 0$$

$$root_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a}$$