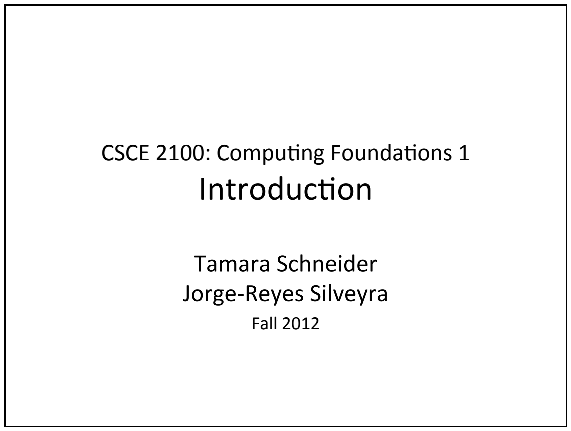
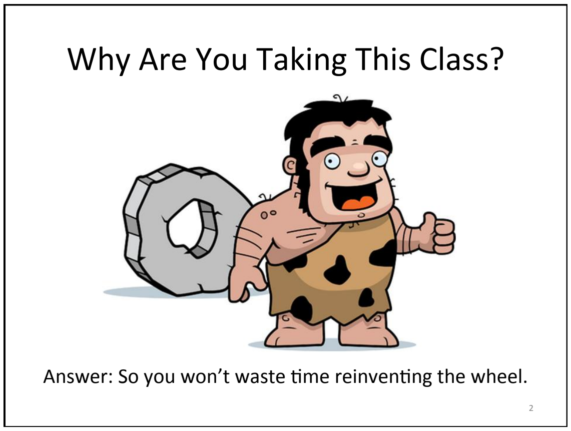


CSCE 2100: Computing Foundations 1

Introduction

Tamara Schneider
Jorge-Reyes Silveyra
Fall 2012

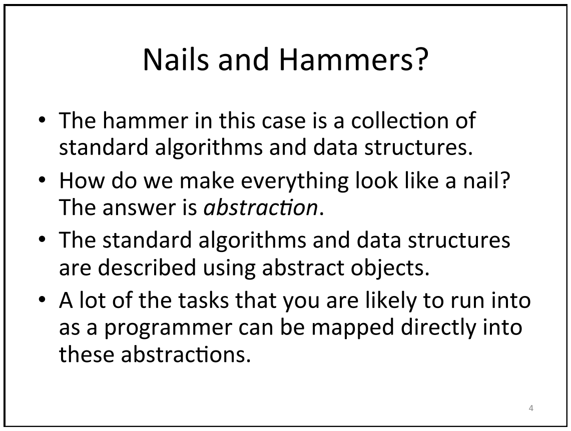


Why Are You Taking This Class?

Answer: So you won't waste time reinventing the wheel.

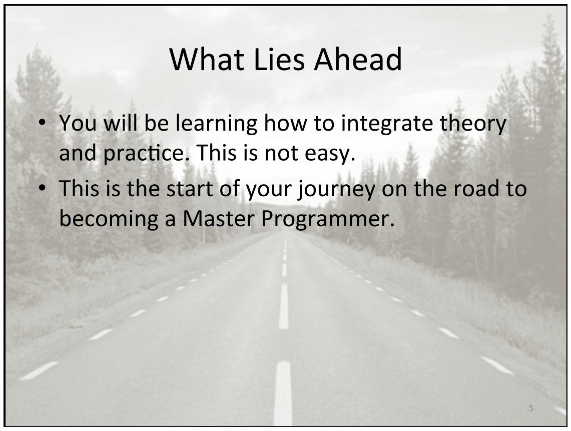
You Know What They Say

When you have a hammer, everything looks like a nail.



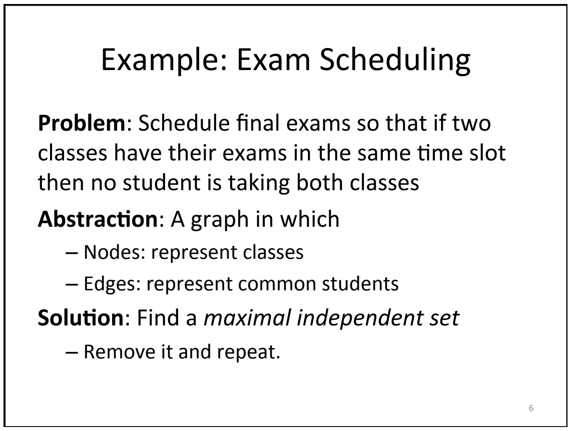
Nails and Hammers?

- The hammer in this case is a collection of standard algorithms and data structures.
- How do we make everything look like a nail? The answer is *abstraction*.
- The standard algorithms and data structures are described using abstract objects.
- A lot of the tasks that you are likely to run into as a programmer can be mapped directly into these abstractions.



What Lies Ahead

- You will be learning how to integrate theory and practice. This is not easy.
- This is the start of your journey on the road to becoming a Master Programmer.



Example: Exam Scheduling

Problem: Schedule final exams so that if two classes have their exams in the same time slot then no student is taking both classes

Abstraction: A graph in which

- Nodes: represent classes
- Edges: represent common students

Solution: Find a *maximal independent set*

- Remove it and repeat.

Example

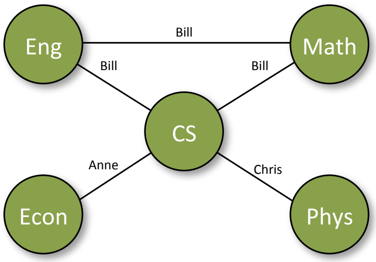
Suppose we have 5 classes to schedule:

1. English
2. Math
3. CS
4. Economics
5. Physics

These are the conflicts:

	Anne	Bill	Chris
English		×	
Math		×	
CS	×	×	×
Econ	×		
Physics			×

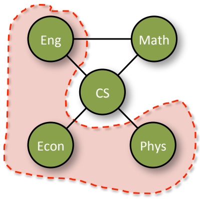
Course Conflict Graph



Definitions

- **Independent set:** A set of nodes that have no edges to other nodes in the set.
- **Maximal independent set:** An independent set to which no additional node can be added.

A Possible Schedule



Maximal Independent Sets:

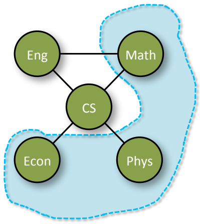
{Eng, Econ, Physics}

{CS}

{Math}

Time	Exams
1	Eng, Econ, Phy
2	CS
3	Math

A Possible Schedule



Maximal Independent Sets:

{Math, Econ, Physics}

{CS}

{English}

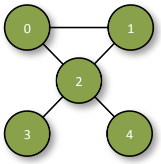
Time	Exams
1	Math, Econ, Phy
2	CS
3	English

Questions

- Does this method always produce a feasible solution?
- Does it always produce a result with the smallest number of time slots?
- Can this abstraction cause any practical problems?
- Can you implement a solution to this problem using techniques you already know?

Implementation

Two-dimensional array (incidence matrix)



	0	1	2	3	4
0	1	1	1	0	0
1	1	1	1	0	0
2	1	1	1	1	1
3	0	0	1	1	0
4	0	0	1	0	1

Data Models

- Abstractions to describe problems
- Examples: trees, lists, sets, relations, finite automata, grammars, logic
- Static and dynamic aspects
  - type system and operations
- Data models in programming languages
  - In C++: Integers, floating-point numbers, structures, pointers, ...

Data Structures

- Data structures represent data models in programming languages.
- They are not explicitly part of the language.
- They are not the same for all languages, as the data models supported by programming languages vary.

Algorithms

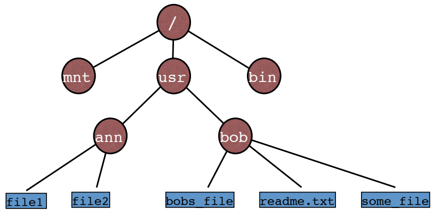
- Sequence of instructions to solve a problem.
- Often described in *pseudo-code*.
- Examples: Sorting, searching, scheduling.
- Important properties: simplicity, running time.

Data Models of Programming Languages

- Data is stored in “boxes”.
- Boxes have types, names, and contain data objects. For example, `int x = 5;` refers to a box of type integer with name `x` containing the data object 5.
- Static: data types (integers, characters, arrays,...)
- Dynamic: arithmetic operations, accessing operations, dereferencing,...

Data Models of System Software

Data is stored in files which are organized in directories.



Data Models of System Software

Processes are executions of programs and can be concatenated

```
 ps -u username | grep gnome | head -2
```

Data Models of Application Software

Software applications have their own data models. For example, a text editor:

- Text strings
- Lines
- Editing operations
  - Insertion, deletion
- Search

Converting Between Number Systems

How do you convert 5_{10} to the binary system?

5	:	2	=	2	R	1
2	:	2	=	1	R	0
1	:	2	=	$\boxed{0}$	R	1

$5_{10} = 101_2$

Converting Between Number Systems

How do you convert 12_{10} to the binary system?

12	:	2	=	6	R	0
6	:	2	=	3	R	0
3	:	2	=	1	R	1
1	:	2	=	$\boxed{0}$	R	1

$12_{10} = 1100_2$

Converting Between Number Systems

How about converting 13_{10} to the tertiary system?

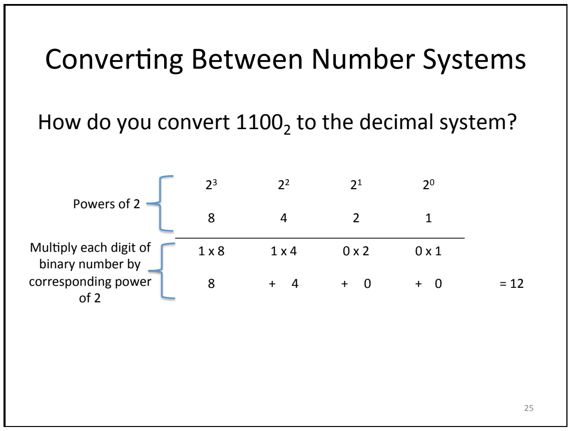
14	:	3	=	4	R	2
4	:	3	=	1	R	1
1	:	3	=	$\boxed{0}$	R	1

$12_{10} = 112_3$

Converting Between Number Systems

How do you convert 1011_2 to the decimal system?

Powers of 2	2^3	2^2	2^1	2^0	
	8	4	2	1	
Multiply each digit of binary number by corresponding power of 2	1 x 8	0 x 4	1 x 2	1 x 1	
	8	+ 0	+ 2	+ 1	= 11

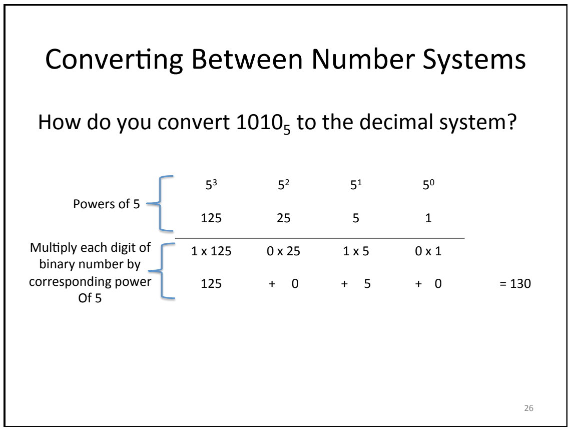


Converting Between Number Systems

How do you convert 1100_2 to the decimal system?

Powers of 2	2^3	2^2	2^1	2^0	
	8	4	2	1	
Multiply each digit of binary number by corresponding power of 2	1 x 8	1 x 4	0 x 2	0 x 1	
	8	+ 4	+ 0	+ 0	= 12

25

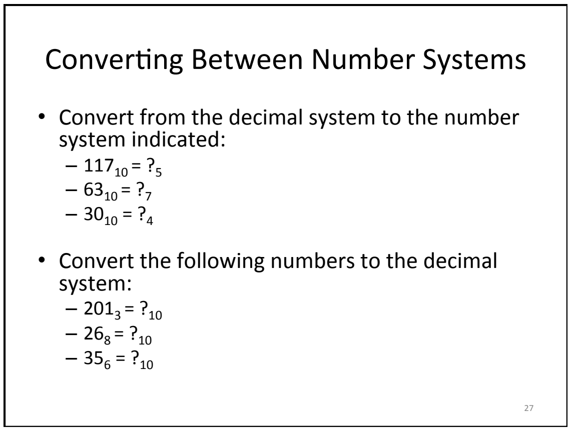


Converting Between Number Systems

How do you convert 1010_5 to the decimal system?

Powers of 5	5^3	5^2	5^1	5^0	
	125	25	5	1	
Multiply each digit of binary number by corresponding power Of 5	1 x 125	0 x 25	1 x 5	0 x 1	
	125	+ 0	+ 5	+ 0	= 130

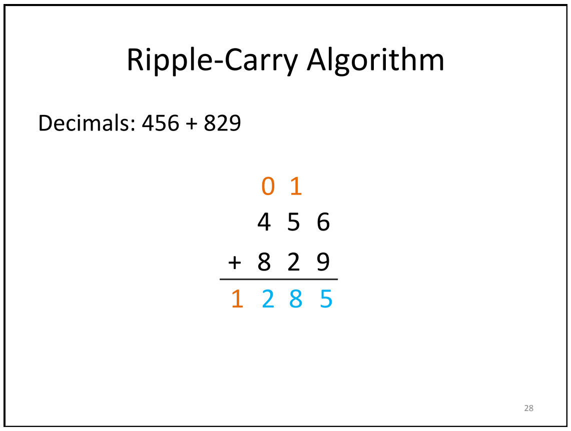
26



Converting Between Number Systems

- Convert from the decimal system to the number system indicated:
  - $117_{10} = ?_5$
  - $63_{10} = ?_7$
  - $30_{10} = ?_4$
- Convert the following numbers to the decimal system:
  - $201_3 = ?_{10}$
  - $26_8 = ?_{10}$
  - $35_6 = ?_{10}$

27

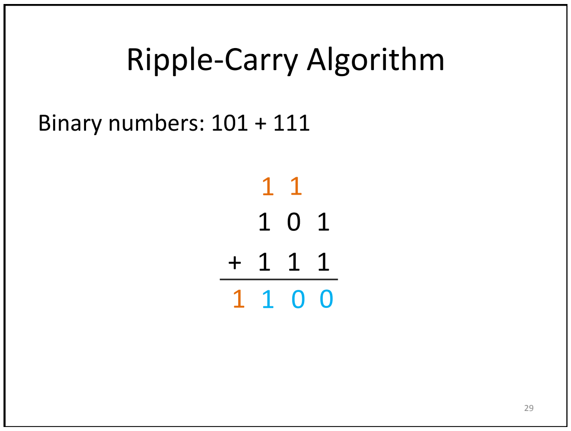


Ripple-Carry Algorithm

Decimals: 456 + 829

$$\begin{array}{ccccc} & & 0 & 1 & \\ & & 4 & 5 & 6 \\ + & & 8 & 2 & 9 \\ \hline & 1 & 2 & 8 & 5 \end{array}$$

28

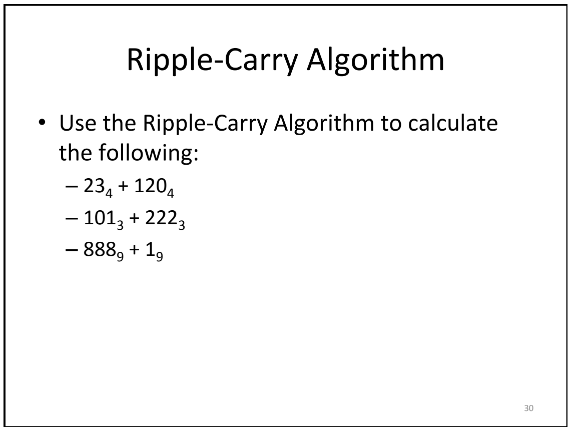


Ripple-Carry Algorithm

Binary numbers: 101 + 111

$$\begin{array}{cccc} & 1 & 1 & \\ & 1 & 0 & 1 \\ + & 1 & 1 & 1 \\ \hline 1 & 1 & 0 & 0 \end{array}$$

29



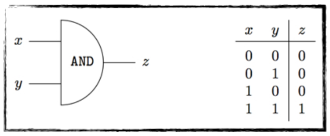
Ripple-Carry Algorithm

- Use the Ripple-Carry Algorithm to calculate the following:
  - $23_4 + 120_4$
  - $101_3 + 222_3$
  - $888_9 + 1_9$

30

Data Models of Circuits

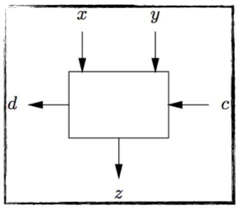
Propositional logic



x	y	z
0	0	0
0	1	0
1	0	0
1	1	1

Data Models of Circuits

One-bit adder



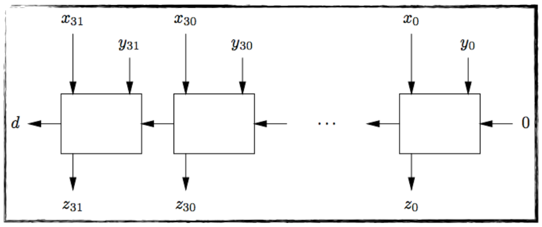
Input: x, y, carry-in bit c

Output: sum bit z, carry-out bit d

x	y	c	d	z
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Data Models of Circuits

Ripple-Carry Adder



Binary Subtraction: 2’s Complement

- $5_{10} - 4_{10} = 101_2 - 100_2$
- 1’s complement of the negative term is 011.
- Add terms 101 and 1’s complement of 100.
- If there is a carry-out, add it to the result.

Binary Subtraction: 2’s Complement

```
    1 1
    1 0 1
+   0 0 0
---------
    0 0 0
      + 1
---------
        1
```

Decimal Subtraction: 9’s Complement

102_{10} - 052_{10}

- 9’s complement of the negative term is 947.
- Add terms 102 and 9’s complement of 52.
- If there is a carry-out, add it to the result.

Decimal Subtraction: 9’s Complement

```
   0 0
   1 0 2
 + 9 4 7
 -------
   0 4 9
      +1
 -------
     5 0
```

Ripple-Carry Algorithm

- Use the Ripple-Carry Algorithm to calculate the following:
  - 120_4 - 23_4
  - 222_3 - 101_3
  - 888_9 - 1_9

Lists

Lists enumerate elements of a specific data type

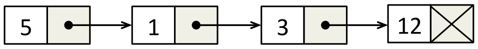
- List of integers: (5, 1, 3, 12)
- List of strings: (“dog”, “cat”, “horse”)

The concept of lists is an *abstraction.*

- Examples of list implementations (data structures): linked lists, array list.
- Some languages have lists as part of their data model, e.g. Lisp, Prolog.

Linked List

Represent (5, 1, 3, 12) as linked list.



Represent abstract lists with a C `struct`.

```
typedef struct cell *list
struct cell{
  int element;
  struct list next;
};
```

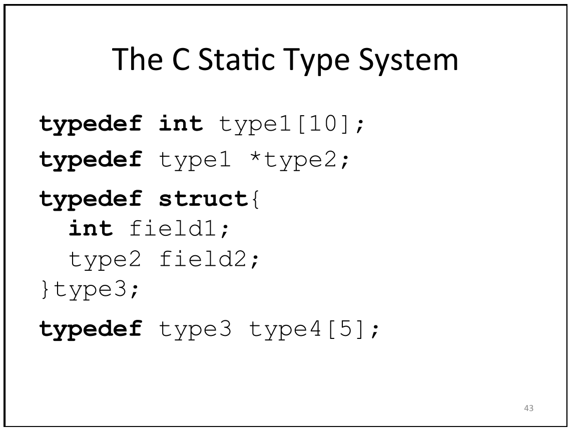
The C Static Type System

Basic types:

- characters
- integers
- floating-point numbers
- enumerations

Type formation rules:

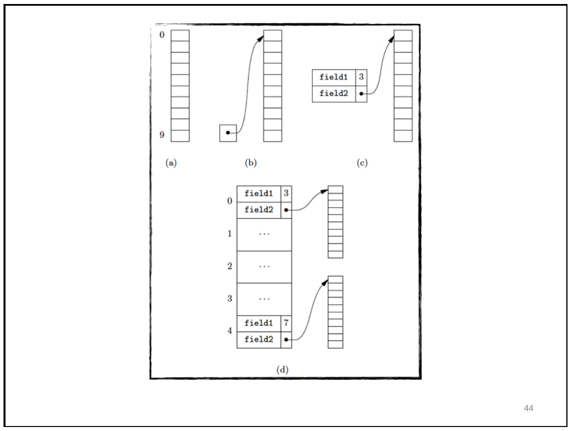
- array types
- structure types
- union types
- pointer types



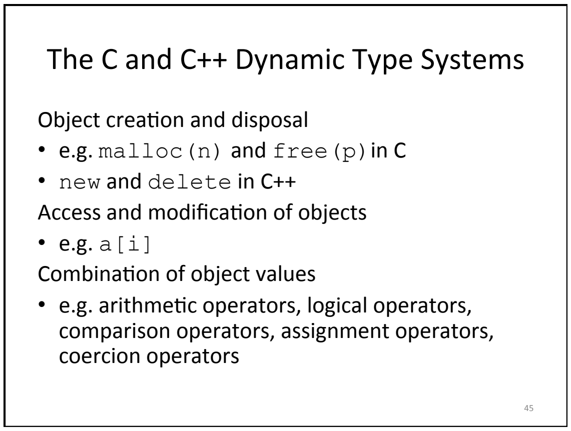
The C Static Type System

```
typedef int type1[10];
typedef type1 *type2;
typedef struct{
  int field1;
  type2 field2;
}type3;
typedef type3 type4[5];
```

43



44



The C and C++ Dynamic Type Systems

Object creation and disposal

- e.g. `malloc(n)` and `free(p)` in C
- `new` and `delete` in C++

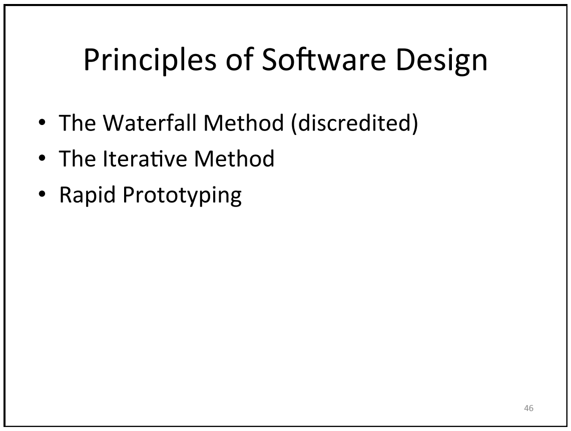
Access and modification of objects

- e.g. `a[i]`

Combination of object values

- e.g. arithmetic operators, logical operators, comparison operators, assignment operators, coercion operators

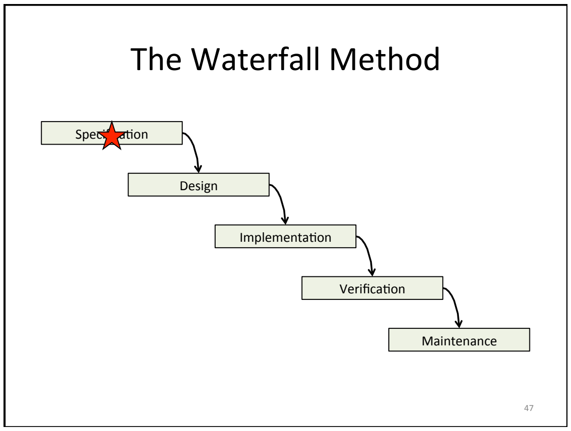
45



Principles of Software Design

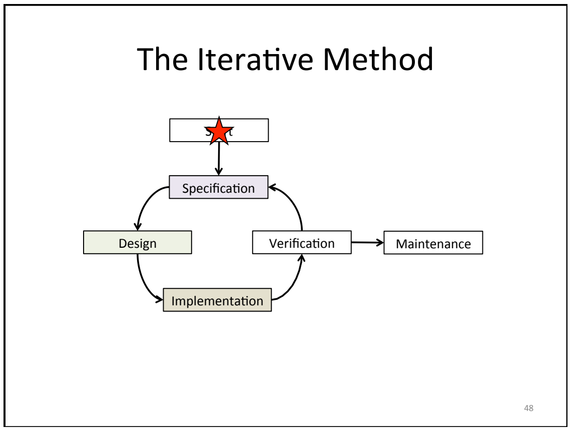
- The Waterfall Method (discredited)
- The Iterative Method
- Rapid Prototyping

46



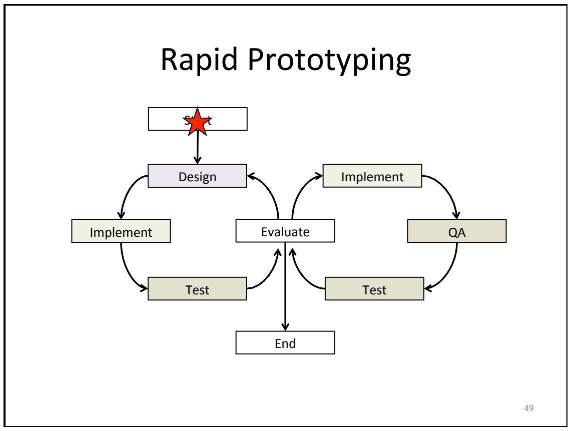
The Waterfall Method

47



The Iterative Method

48



Rapid Prototyping

Start

Design

Implement

Evaluate

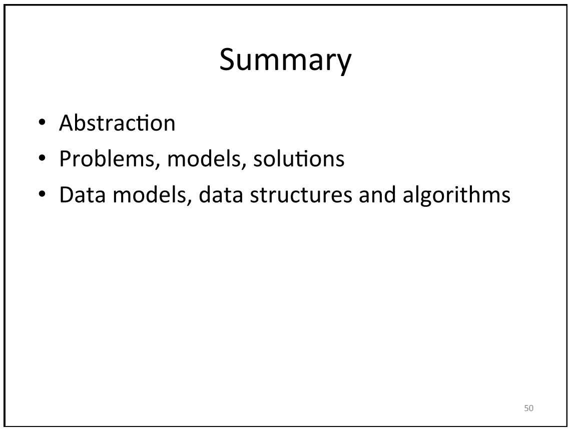
Implement

QA

Test

Test

End



Summary

- Abstraction
- Problems, models, solutions
- Data models, data structures and algorithms