Introduction to problem solving in CS

Computer Applications in
Power Systems – Advance course

EH2750

Dr. Arshad Saleem

10-Sep-12

outline

- Prt I
  - Lecture objectives
  - revisting Architecture
  - System development lifecycle
  - Problem solving steps in CS
- Part II
  - Introduction to programming languages and OOPs in JAVA

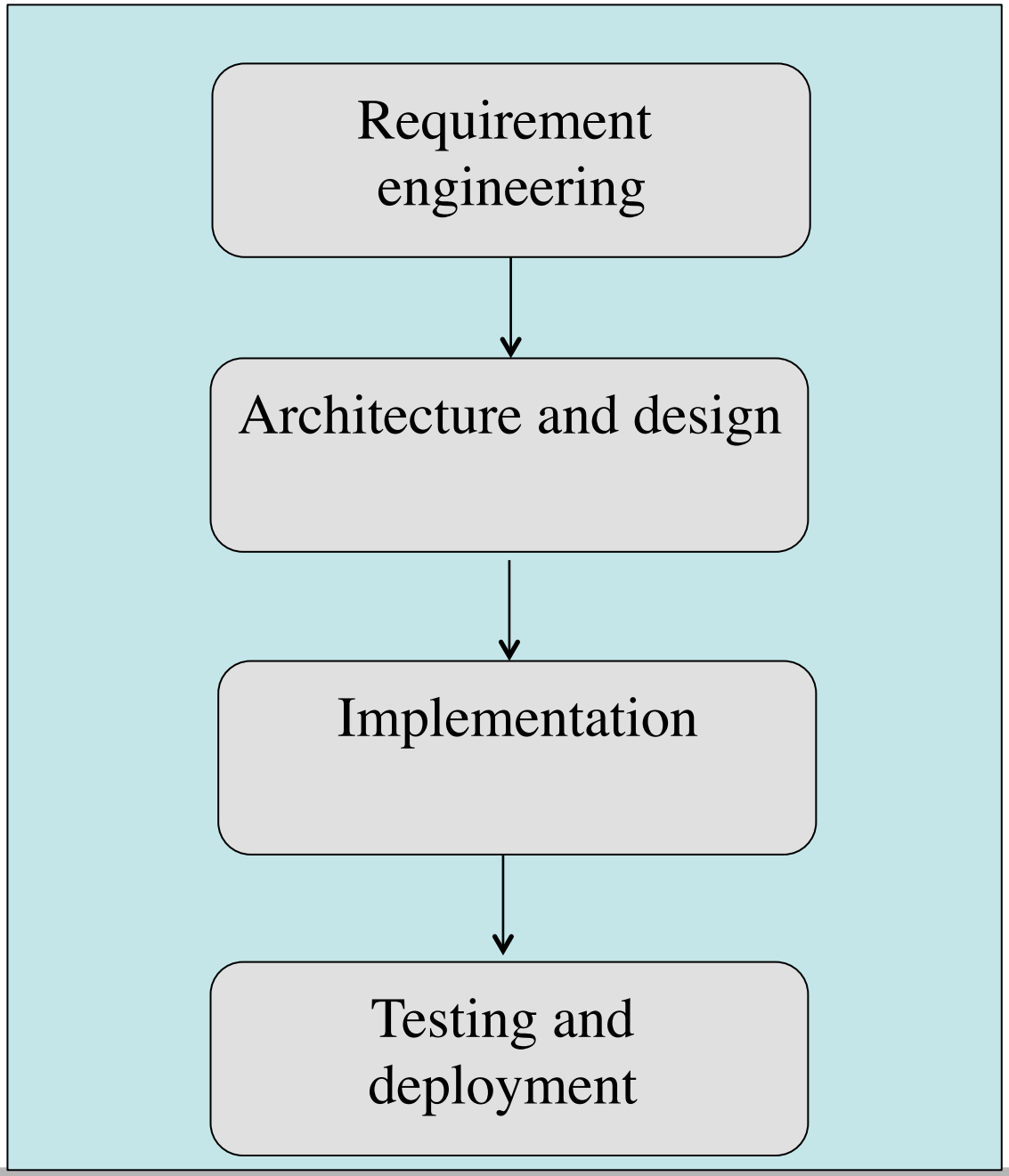
Lecture objectives

- Make you start thinking in terms of CS based problem solving
  - From component to system level approach
  - From signals to actors and functions
  - From block diagram to architecture and design
  - From matLAB to Java

What is Architecture?

- Description of system in terms of its sub-compoents, functions, internal interactions and interfaces
  - UML diagrams and Reference architecture are kind of templates
  - Architecture is trasformed into a implementation
  - Enables reusability, changeability, interoperability
  - We are mostly intereste in system and control architecture

System development life cycle



Problem solving steps in CS

- 1. Understand the Problem
- 2. Formulate a Model
- 3. Develop an Algorithm
- 4. Write the Program
- 5. Test the Program and evaluate the solution

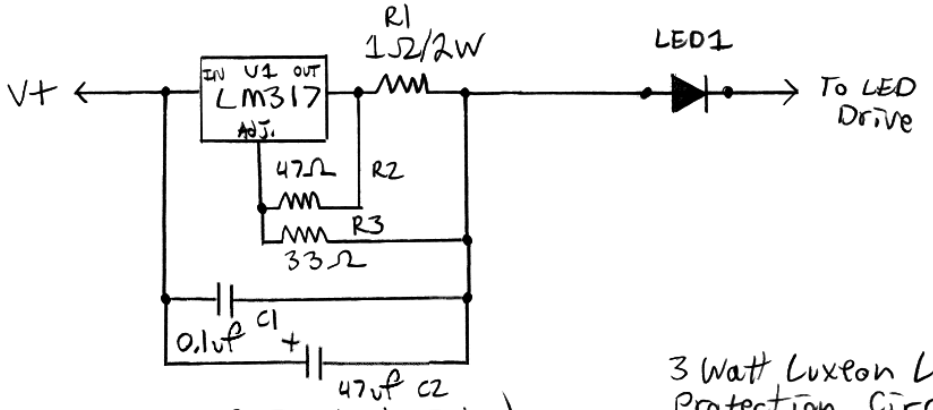
Understand the Problem

- Understand the objectives
- Understand and identify entities and functions
- Understand inputs and outputs
- Do i have everyting i need
- What is missing
- What do I want the result to look like … text, a picture, a graph, control signal, setpoint … ?

What it will be in case of overcurrent relay operation?

Formulate a Model

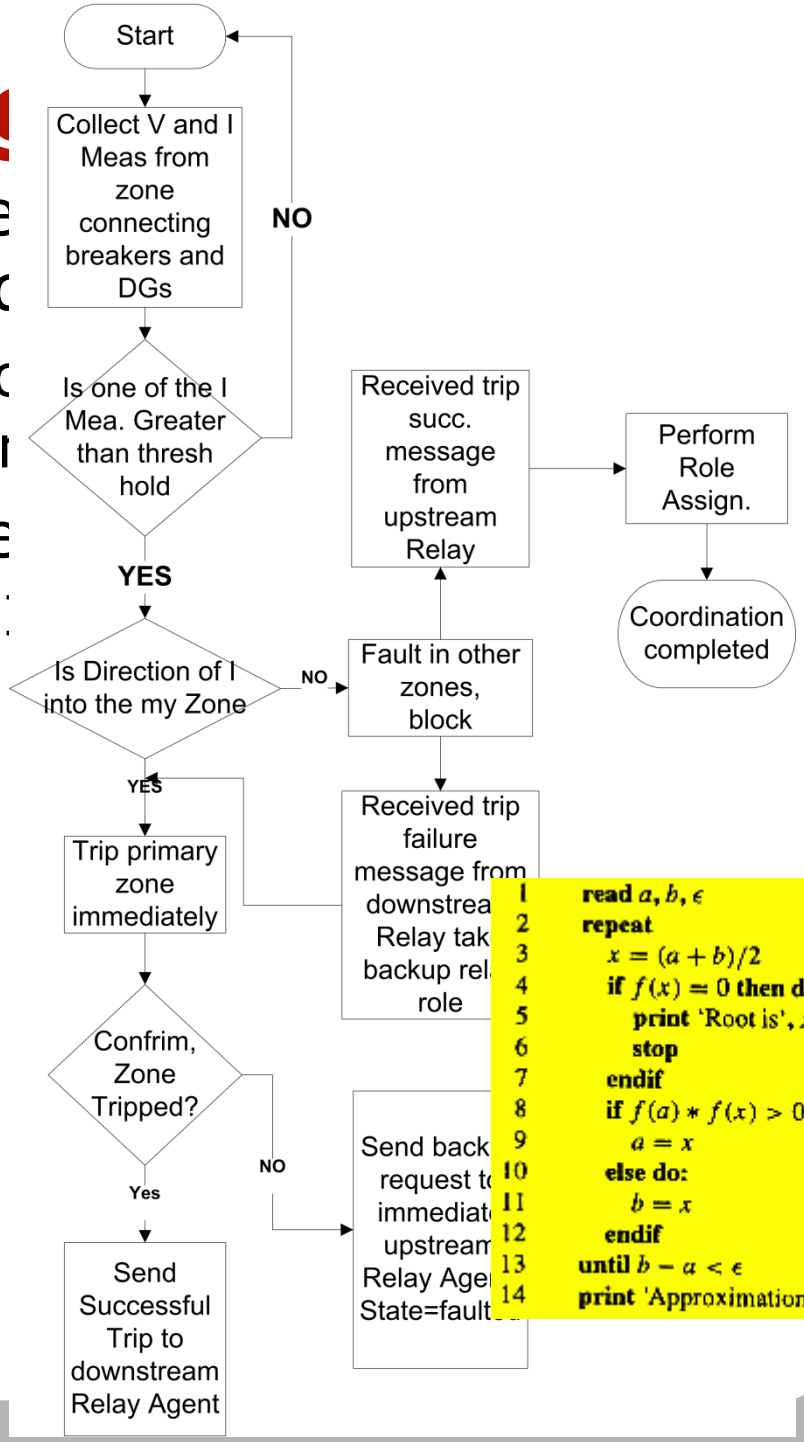
- Understand the processing/computation part of the problem
- break down into smaller problems if possible
- Formulate the problem into mathematial formulation
  - Use some existing or design new one



Develop an Alg

- come up with a precise
want the computer to c
  - Sequence of instructic
for solving the proble
- Two commonly used re
algorithm is by using (
(2) ***flow charts***
- Sequence, condition

and repition

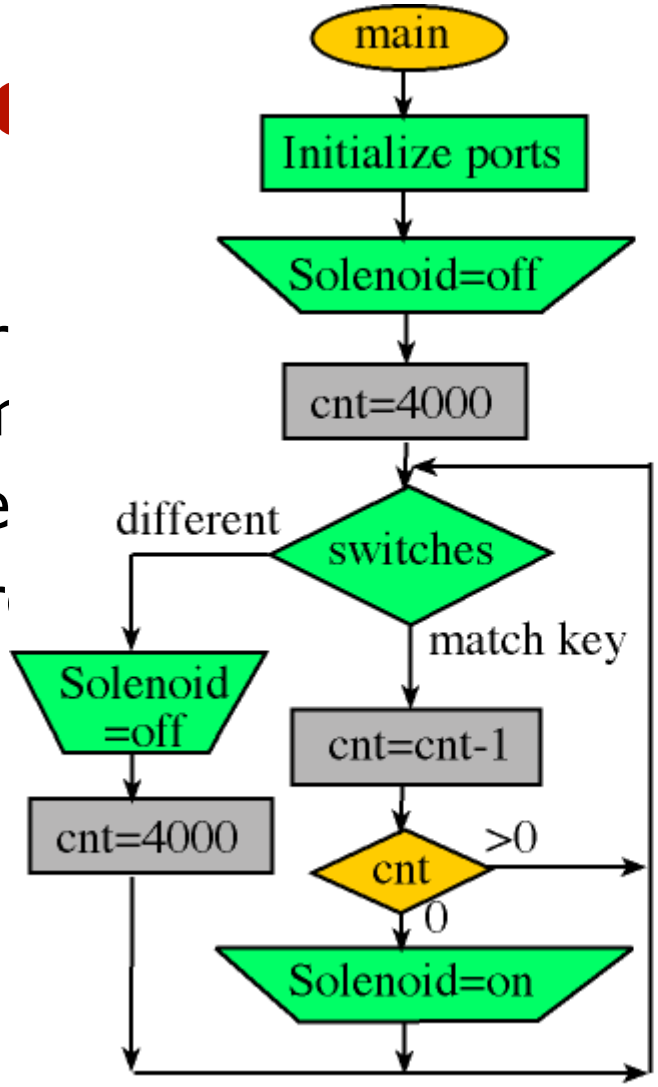



```
1   read a, b, ϵ
2   repeat
3     x = (a + b)/2
4     if f(x) = 0 then do:
5       print 'Root is', x
6       stop
7     endif
8     if f(a) * f(x) > 0 then do:
9       a = x
10    else do:
11      b = x
12    endif
13  until b − a < ϵ
14  print 'Approximation to root is', x
```

Write the Pr

- Transform the algor written in a program
- Follow the language
- Implement hardwar connections etc
- This is much easier

and clean if previous steps are done properly

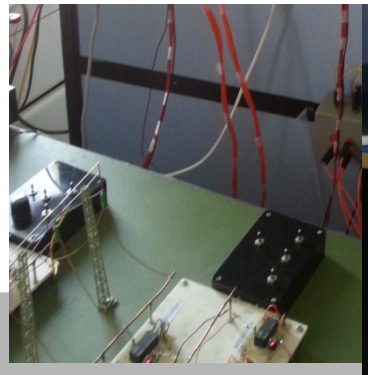


Pseudo Code

```
1) initialize ports
     PA6-PA0 inputs
     PA7 output
2) turn off solenoid
3) set counter to 4000
4) repeat indefinitely
   if switch matches ke
     a) decrement coun
     b) if counter is zer
        turn on solenoid
   otherwise
     a) turn off solenoi
     b) set counter to 4
```

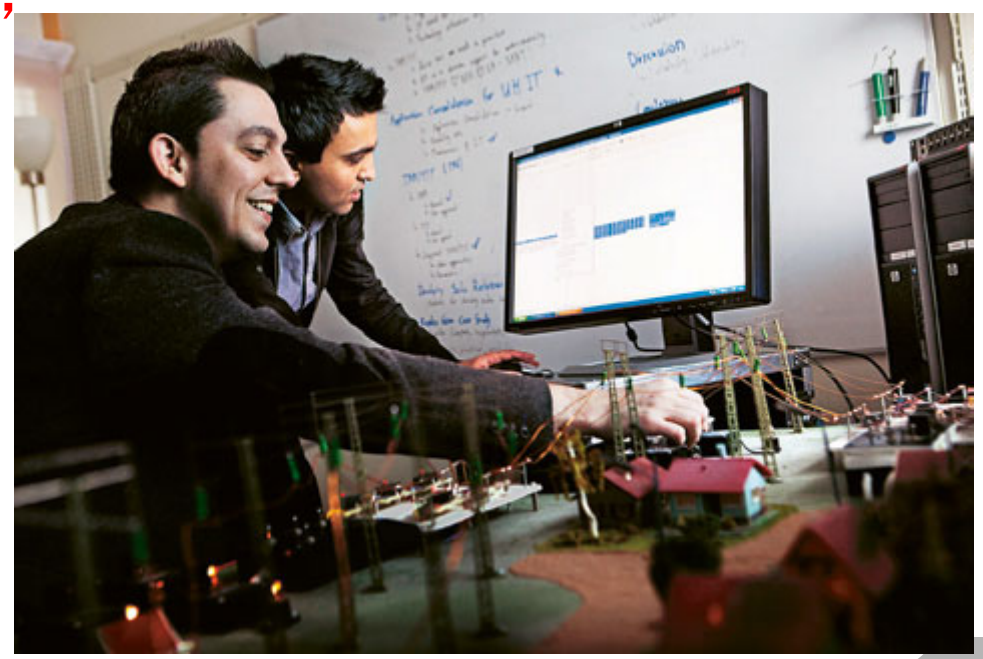
C Code

```
DDRA=0x80;
PORTA=0; cnt=4000;
while(1){
  if((PORTA&0x7F==k
    if((--cnt)==0)
      PORTA|=0x80;}
  else{
    PORTA=0; cnt=40
```



Testing and result evaluation

- make sure that it solves the problem that it was intended to solve
- All possible conditions have been tested
- Results are presneted properly
- Joy of ’It Works’



Problem solving steps in CS

- 1. Understand the Problem
- 2. Formulate a Model
- 3. Develop an Algorithm
- 4. Write the Program
- 5. Test the Program and evaluate the solution

System development life cycle