

LECTURE 01

PROGRAM DESIGN

Problem Solving - A general Perspective

You are back home from a leisure stroll in the park, and discover that you have dropped your wallet that is of great sentimental value to you. The park is a rectangular block of 2kmx1km.

How do you retrieve your lost wallet?

Nature of Problems

Problems can range from

- Simple to complex
- Mathematical to Philosophical



In computing we deal with “algorithmic problems”!!



Returning to the lost-wallet problem,

- Problem statement incomplete!!
- Were Assumptions made?

Prior to solving the problem,

- Assumptions need to be made with a basis
- Irrelevant information need to be discarded
- Remove all ambiguities prior to understanding the problem.

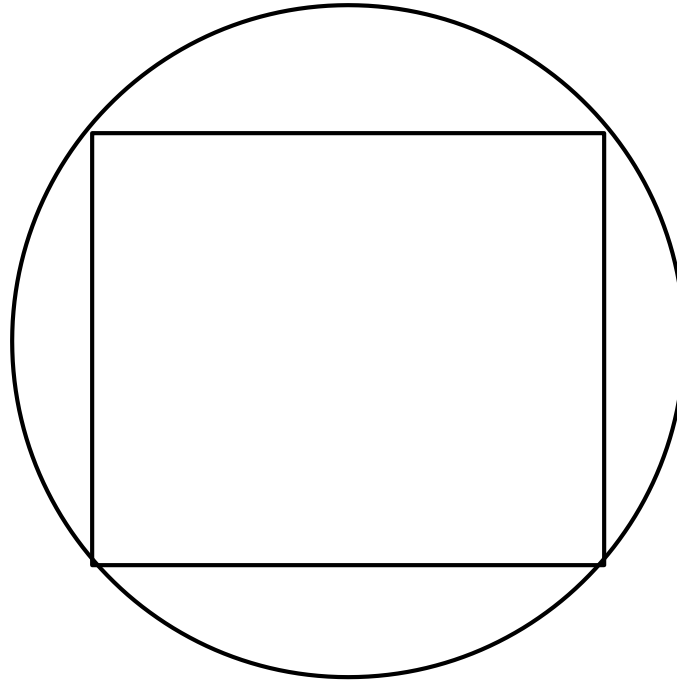
Problem - reformulated

Suppose the wallet was lost in a rectangular park by 2km by 1km how would you recover the wallet?



Problem 2: Mathematical

A square, where each side has length $2a$, is inscribed in a circle. What is the area of the circle?



Problem 3: Logical

- You are to ship **a wolf, a sheep and a cabbage** across the river.

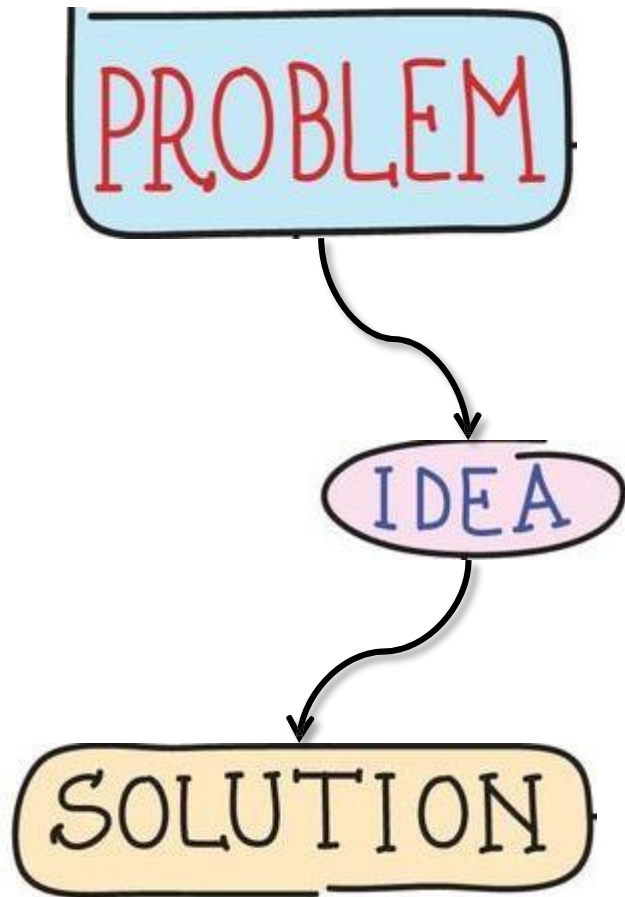
The boat can only hold the weight of two: you plus another item. You are the only one who can row the boat. The wolf must not be left with the sheep unsupervised, or the wolf will eat the sheep. Neither should the sheep be left alone with the cabbage. How do you fetch them over to the other side of the river?

Computational Approach for Problem Solving



In computing we focus on problems, where the solutions are expressible in the form of an algorithm.

Steps in Problem Solving



Analyze the
problem

Outline the Solution

Develop
Algorithm

Steps in Problem Solving

1. Analyze the Problem

- Does user require User Interaction?
- Does the program manipulate data?
- Identify Inputs & Outputs

2. Outline the Solution

- Major processing Steps
- Control Structures (eg: conditions, loops)

3. Develop Algorithm

- Develop the outline into an algorithm



Write the **steps** to log in to your Facebook account.



1. Go to www.facebook.com
2. Enter email id & password
3. Click on Login button



A set of steps defining how a task is performed

What is an Algorithm?

- Informally: “An algorithm is a set of steps that define how a task is performed.”

- Formally: “An algorithm is an ordered set of unambiguous executable steps, defining a terminating process.”
 - Ordered set of steps: structure!
 - Executable steps: doable!
 - Unambiguous steps: Well defined instructions!
 - Terminating: must have an end!

Tools for Representing Algorithms

1. Flowchart
2. Pseudocode
3. Structure Chart

Introduction to Algorithms

Problem Solving Phase

1. Define the problem
 - The inputs
 - The outputs
 - The processing steps to produce the required outputs
2. Outline the solution
 - The major processing steps involved
 - The subtask (if any)
 - The user interface (if any)
 - The major control structures (e.g. Repetition loops)
 - The major variables and record structures
 - The mainline logic
3. Develop the outline into an algorithm
4. Test the algorithm for correctness (deskchecking)

Implementation Phase

4. Code the algorithm into a specific programming language
5. Run the program on the computer
6. Document and maintain the program

Example 1 : Add three numbers

A program is required to read three numbers add them together and print their total.

Outline the Solution-Defining Diagram

- Underline the nouns and adjectives → establish the input, output component and any object that are required.

A program is required to read three numbers add them together and print their total.

Defining Diagram

Input	Processing	Output
number1		total
number2		
number3		

- Underline the verbs and adverbs → establish the actions required.

A program is required to read three numbers add them together and print their total.

Defining diagram

Input	Processing	Output
number1	Read three numbers	total
number2	Calculate total by adding three numbers	
number3	Print total	

Solution Algorithm

1. Read number1, number2, number3
2. Calculate total = number1 + number2 + number3
3. Print total

Example 2 - Find average temperature

A program is required to prompt the terminal operator for the maximum and minimum temperature readings on a particular day accept those readings as integers and calculate and display to the screen the average temperature calculated by $(\text{maximum temperature} + \text{minimum temperature})/2$.

Solution

A program is required to prompt the terminal operator for the **maximum and minimum temperature readings** on a particular day accept those readings as integers and calculate and display to the screen the **average temperature** calculated by $(\text{maximum temperature} + \text{minimum temperature})/2$.

Defining diagram

Input	Processing	Output
max_temp min_temp	Prompt for temperatures Read temperatures calculate average temperature display average temperature	avg_tem p

□ Solution Algorithm

1. Prompt operator for max_temp, min_temp
2. Get max_temp, min_temp
3. $\text{avg_temp} = (\text{max_temp} + \text{min_temp})/2$
4. Output avg_temp to the screen

Homework: Compute mowing time

A program is required to read from the screen the length and width of a rectangular house block and the length and width of the rectangular house that has been built on the block.

The algorithm should then compute and display the mowing time required to cut the grass around the house at the rate of two square meters per minute.



Q&A ?