Problem Solving

- Implementing a solution by breaking the bigger problem into smaller problems.

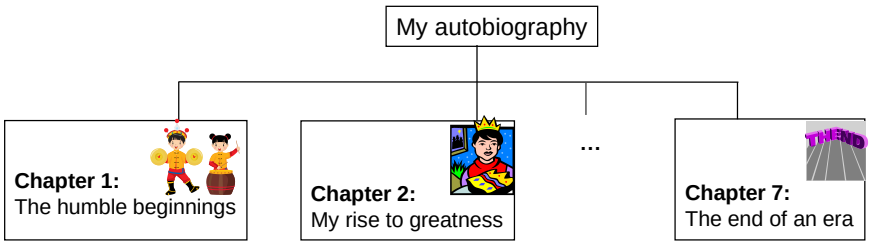
"Problem Solving"

- When faced with a daunting problem (too big, too complex so you cannot start) you have been taught to break things down into parts.
- This is referred to as "top down decomposition"

James Tam

Top Down Design

1. Start by outlining the major parts (structure)



2. Then implement the solution for each part

Chapter 1: The humble beginnings

It all started ten and one score years ago with a log-shaped computer work station…

Image copyright unknown

Top Down Design: Starting Rule Of Thumb

- General approach (any task):
  - Break the problem down into 3 to 5 smaller parts e.g. book chapters.
  - If any of those parts are too large and complex they may be decomposed into smaller parts e.g. sections of a book.
- Writing a program:
  - Programs/software can have 'bugs' (errors) so programmers need to think about what task that each part of the program should complete and then run tasks to see if the program is correct.
    - Example results:
      - A computed value.
      - Showing an image if the option is selected by the user.
      - Saving a file in a certain form if certain conditions are met.
  - Testing can involve running possible options (cases) that might produce an unexpected value or effect e.g. when showing the image what if the user selecting some other options before selecting the option to display the image.

James Tam

Example Problem

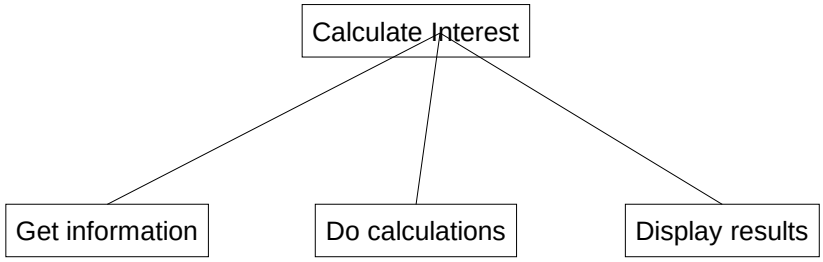
- Design a program that will perform a simple interest calculation.
- The program should prompt the user for the appropriate values, perform the calculation and display the values onscreen.

Example Problem

- Design a program that will perform a simple interest calculation.
- The program should prompt the user for the appropriate values, perform the calculation and display the values onscreen.

Example Problem

- Design a program that will perform a simple interest calculation.
- The program should prompt the user for the appropriate values, perform the calculation and display the values onscreen.



Discussion: Cases To Test

- Here's something to get you started:
  - Principle is greater than zero.
  - Interest is greater than zero.
  - Time unit is greater than zero.
  - Example: $100 invested at a yearly return of 1% for 3 years.
- What other cases could be tested:

James Tam

JT: Look Ahead

- You will likely see much more on the top down approach for decomposition in a later section: “decomposition using functions”.

James Tam

Algorithms

- This is not necessarily mathematical so don’t get all tense! ;-)
- Definition:
  - A finite sequence of steps that clearly provides the correct solution to a problem.
- It may be the output of applying the top down approach (or some other approach).
  - Example:
    - Get the principle, rate and time.
    - Compute the simple interest and the current amount earned.
    - Display the original 3 values and the 2 computed values.
- Language used:
  - A human language
  - Properly specified the algorithm can be translated to a programming language (such as python).

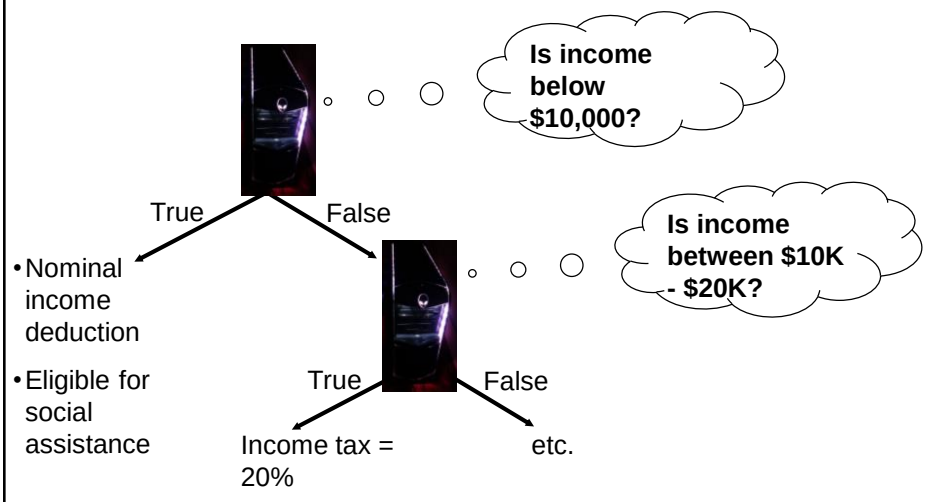
James Tam

Algorithms

- This is not necessarily mathematical so don't get all tense! ;-)
- Definition:
  - A finite sequence of steps that clearly provides the correct solution to a problem.
- It may be the output of applying the top down approach (or some other approach).
  - Example:
    - Get the principle, rate and time.
    - Compute the simple interest and the current amount earned.
    - Display the original 3 values and the 2 computed values.
- Language used:
  - A human language

James Tam

Example Algorithm



James Tam

Programming

- Properly specified the algorithm can be translated to a programming language (such as python).
- Programming definition:
  - Creating a computer program by translating an algorithm into a programming language.

James Tam

Program Interpreter.

- A special computer program that performs a literal line-by-line translation from the programming language to machine language.
- Human languages are specified the rules known as *grammar*.
  - Breaking these rules may still allow another person to interpret the meaning.
- Programming languages are specified the rules known as the *syntax*.
  - Violate the syntax of a programming language in any way (e.g. missed a closing bracket, there's a small typo etc.) and translation will fail.
    - Example language python, basic: partial translation may still occur (program is translated and runs up to the point just before the error).
    - Example languages Java, C, C++: no translation occurs i.e. no part of the program will run!
    - **JT: You must carefully learn & apply the syntax of each language** (in my notes you will see the heading 'format' to illustrate the required structure).

James Tam