

Method Abstraction and Stepwise Refinement

Ch. 6 Concepts, pg. 206 - 213

Method abstraction

- **Method abstraction** is the separation of a method's use from its implementation.
- The user of a method (programmer) can use it without knowing how it is implemented.
 - The method is treated as a "black box."
- Examples?
 - `Math.random()`
 - `System.out.print(String text)`
 - `Math.round(int a)`
 - `Math.max(int a, int b)`

Stepwise refinement

- Divide problems into smaller, more manageable subproblems
- "Divide and conquer" approach
- This approach can be reapplied to those subproblems

Top-down design

- Worrying about a problem's details at the beginning of addressing it can lead to confusion or can be overwhelming.
- We can use method abstraction to structure a more productive approach.
- Think about the subproblems and *what they achieve*, not on how to code them.
- Let's look at an example...

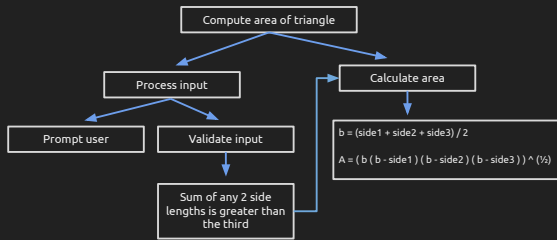
But first, a video!

<https://www.youtube.com/watch?v=PRcTIFzk-4k>

Problem: computing triangle area

- Prompt the user for three side lengths of a triangle.
- Output an error message if the triangle is invalid.
- Compute the area of the triangle using the following formulae.
 - $b = (\text{side1} + \text{side2} + \text{side3}) / 2$
 - $A = (b - \text{side1}) (b - \text{side2}) (b - \text{side3}) ^ {1/2}$

Top-down design



Now, to code it!

But first, another video!

<https://www.youtube.com/watch?v=6WuVZ2QUvGE>
