

Table of contents

- [1.](#)
 - [2.](#)
 - [3.](#)
 - [4.](#)
 - [5.](#)
 - [6.](#)
 - [7.](#)
 - [8.](#)
 - [9.](#)
 - [10.](#)
 - [11.](#)
 - [12.](#)
 - [13.](#)
 - [14.](#)
 - [15.](#)
 - [16.](#)
 - [17.](#)
 - [18.](#)
 - [19.](#)
 - [20.](#)
 - [21.](#)
 - [22.](#)
 - [23.](#)
-

1. Top Down Design

- First, design the top level of each stage of the algorithm. focusing on what they'll do rather than how (code) they'll do it.
- Then dive into each stage focussing on what
- When it no longer makes sense to break the program into smaller functions, then focus on how (code) each stage will complete its task.

([go to top](#))

2. Chunks of Functions

- Instead of writing one longer algo. create functions where they can be applied and then write the algo by calling the functions.

([go to top](#))

3. Unit Testing

- Test each function block before putting them all together?

([go to top](#))

4. Prototyping

- Rather than taking the entire problem and proceeding through specification, design, implementation, and testing, we first design, implement, and test a prototype. Then new features are designed, implemented, and tested. We make many mini-cycles through the development process as the prototype is incrementally expanded into the final program.

([go to top](#))

5. Title

([go to top](#))

6. Title

([go to top](#))

7. Title

([go to top](#))

8. Title

([go to top](#))

9. Title

([go to top](#))

10. Title

([go to top](#))

11. Title

([go to top](#))

12. Title

([go to top](#))

13. Title

([go to top](#))

14. Title

([go to top](#))

15. Title

([go to top](#))

16. Title

([go to top](#))

17. Title

([go to top](#))

18. Title

([go to top](#))

19. Title

([go to top](#))

20. Title

([go to top](#))

In []:

In []: