CPT108 Data Structures and Algorithms

Mistakes People do while Learning Programming

Mistake People do while Learning Programming

- Attempting to memorize the code!
  - While some memorization is necessary, it's more important to understand the underlying principles and problem-solving strategies in programming
  - With this approach, you will end up with a lack of understanding and nothing else

Mistake People do while Learning Programming

- Attempting to memorize the code!
  - While some memorization is necessary, it's more important to understand the underlying principles and problem-solving strategies in programming
  - With this approach, you will end up with a lack of understanding and nothing else
- Don't read the error message or try to understand them, or not testing your code as your go
  - Sweeping bugs under the carpet

Mistake People do while Learning Programming

- Attempting to memorize the code!
  - While some memorization is necessary, it's more important to understand the underlying principles and problem-solving strategies in programming
  - With this approach, you will end up with a lack of understanding and nothing else
- Don't read the error message or try to understand them, or not testing your code as your go
  - Sweeping bugs under the carpet
- Not practising enough, just wait for other people's answers or the solution on the Internet or generative AI
  - Like any skill, programming requires practice. It's important to write code regularly and work on projects to reinforce your learning
  - Programming can be challenging, and it's important to be *patient* with yourself as you learn new concepts and skills
  - It's okay to ask for help when you're stuck. Whether it's from a mentor, online community, or tutorial, seeking guidance can help you overcome obstacles

Mistake People do while Learning Programming (cont.)

- Not thinking logically (step-by-step, like a computer)
  - You should find out all the (possible) scenarios that may appear
  - You should first try to solve the problem on paper, *NOT* on computer
  - You don't need to write entire code on paper. Just think and write the steps (like pseudocode) for problems, or some images illustrating the flow of the operations under different cases

Mistake People do while Learning Programming (cont.)

- Not thinking logically (step-by-step, like a computer)
  - You should find out all the (possible) scenarios that may appear
  - You should first try to solve the problem on paper, *NOT* on computer
  - You don't need to write entire code on paper. Just think and write the steps (like pseudocode) for problems, or some images illustrating the flow of the operations under different cases
- Overlooking the basics
  - It's crucial to have a solid understanding of the fundamentals of programming before diving into more advanced topics
  - Skipping over basic concepts can lead to confusion later on

Mistake People do while Learning Programming (cont.)

- Not thinking logically (step-by-step, like a computer)
  - You should find out all the (possible) scenarios that may appear
  - You should first try to solve the problem on paper, *NOT* on computer
  - You don't need to write entire code on paper. Just think and write the steps (like pseudocode) for problems, or some images illustrating the flow of the operations under different cases
- Overlooking the basics
  - It's crucial to have a solid understanding of the fundamentals of programming before diving into more advanced topics
  - Skipping over basic concepts can lead to confusion later on
- Leaving gap in your learning

Solution of Lab 2 Recursion - Palindrome

```
public class PalindromeSol {
  private static Pattern pattern = Pattern.compile("[A-Za-z0-9]+");

  private String normalize(final String str) {
    if (null == str || str.isBlank()) return "";
    return str.replaceAll("[\\p{Punct}\\s]", "").toLowerCase();
  }

  // private String normalize(final String str) {
  // Matcher matches = pattern.matcher(str);
  // return matches.results().map(o -> o.group()).collect(Collectors.joining()).toLowerCase();
  // }

  private boolean verifyRecursively(final String str) {
    int len = str.length();
    if (len < 2) return true;
    return str.charAt(0) == str.charAt(len - 1) && verifyRecursively(str.substring(1, len - 1));
  }

  public boolean verifyRecursive(String str) {
    String s = normalize(str);
    if (s.isBlank()) return false;
    return verifyRecursively(s);
  }

  public boolean verifyIterative(String str) {
    String s = normalize(str);
    if (s.isBlank()) return false;
    if (s.length() < 2) return true;
    int len = s.length();
    for (int i = 0; i < s.length() / 2; i++) {
      if (s.charAt(i) != s.charAt(len - i - 1)) return false;
    }
    return true;
  }
}
```

Dos

- Don’t give up
  - It is just part of becoming a programmer or software engineer

Dos

- Don’t give up
  - It is just part of becoming a programmer or software engineer
- Don’t direct type on computer, think twice (or more) before typing the code on computer

Dos

- Don't give up
  - It is just part of becoming a programmer or software engineer
- Don't direct type on computer, think twice (or more) before typing the code on computer
- Don't focus too much on syntax
  - Focus on the *logic* and make refinements to a program, or the algorithm, *before* translating it into programming language

Dos

- Don't give up
  - It is just part of becoming a programmer or software engineer
- Don't direct type on computer, think twice (or more) before typing the code on computer
- Don't focus too much on syntax
  - Focus on the *logic* and make refinements to a program, or the algorithm, *before* translating it into programming language
- Do trace your code

Dos

- Don't give up
  - It is just part of becoming a programmer or software engineer
- Don't direct type on computer, think twice (or more) before typing the code on computer
- Don't focus too much on syntax
  - Focus on the *logic* and make refinements to a program, or the algorithm, *before* translating it into programming language
- Do trace your code
- ***Practice* *makes perfect!***
  - e.g., `https://codingbat.com/java`, `https://exercism.org/`, `https://www.codechef.com/practice/java`, ...

References

Goel, Mallika (2020). *What are the mistakes people do while learning prorgamming?* Online: https://www.quora.com/What-are-the-mistakes-people-do-while-learning-programming.

Hennessy, Jim (2021). *Ten Do's and Dont's for Living and Learning during Code Bootcamp*. Online: https://www.codeplatoon.org/ten-dos-and-donts-for-living-and-learning-during-coding-bootcamp/.