Good Programming

To work program: Design a program -> Write the program in a programming language -> Interpret or compile the program -> Run it and see the results.

**To Work Program**: The elements for Systematic Program Designs are Problem Analysis and Data Definitions, Contract (Signature), Purpose (Effect) statement, Header, Functional Examples, Function Definition, and Testing.

**Write the program in a programming language:** Programming Language consists of Peculiar Syntax, some behavior associated with each syntax (semantics), Libraries, and idioms (specific language usage)

**To interpret or compile the program:** The interpreter takes a program and produces a result/Compiler takes a program and produces a program.

텍스트, 스크린샷, 폰트, 라인이(가) 표시된 사진

자동 생성된 설명

**Designing a good program:** Contact, purpose, tests, header, and body

Test-Driven Development: Write a test case before writing programs, Write the simplest code to pass, and if your code is not sufficient, write more tests and repeat.

Tests are a great source of documents: precise and executable and always in sync. **Benefit**: Keep simple design, incremental progress, and protect

Racket Basics

(operator operand operand) / (define (function-name param1 …) body) (function-name args1 … ) / Booleans and relations e.g) (and (> 4 3) (<= 10 100)) / (define (function-name param1 …) cond [ce1 body1] … [else body])) / Only one basic operation on symbols: symbol=?