Programming Methodology

Vineeth Paleri

National Institute of Technology Calicut

Introduction

Introduction

Goal:

Reliable Software

Introduction

- **Programming began as an art** and even today most people learn only by watching others perform and through habit, with little direction as to the principles involved.
    - Programming requires **conscious application of principles.**

Introduction

- **Students are not being taught how to program;** they are only being taught a programming language.
    - Programming Languages may come and go, but programming is here to stay.

Programming: Principles and Techniques

Programming: Principles and Techniques

- Teaching programming means
  - explaining **problem solving ideas**,
  - teaching **orderly thinking**,
  - getting across a sense of **simplicity**, elegance, and style.

Programming: Principles and Techniques

- There are many **guiding principles** to deal with complex problems:
  - **separation of concerns**
  - **modular design**
    - **procedural abstraction, data abstraction**
  - **structured programming**

Programming: Principles and Techniques

Making the simple complicated is common place; **making the complicated simple, awesomely simple, that's creativity.**

- Charles Mingus

Specification

Specification

- Before attempting to solve a problem make absolutely sure you know what the problem is.
  - If the specifications are **contradictory**, then no program will satisfy it.
  - If the specification is **absolutely ambiguous**, any program will satisfy it.

Specification

- **Example:** Division of two integers to get the quotient and the remainder, by repeated subtraction.

Specification

- **Input:**

  dividend $>=$ 0

  divisor $>$ 0

- **Output:**

  quotient $>=$ 0

  0 $<=$ remainder $<$ divisor

(Note: dividend = quotient * divisor + remainder)

Design

Design: Fundamental Principle

- "**Separation of concerns** is the only available technique for effectively ordering one's thoughts."
  - We know that a program must be **correct** and we can study it from that view point only;
  - We also know that it should be **efficient** and we can study its efficiency on another day.

Design: Independent Modules

- There is only one way to deal complex problems:
  - divide it into **smaller independent modules**

Design: Procedural Abstraction

- **Procedural abstraction** separates the concern of how a procedure would be **used** from the concern of how the procedure would be **implemented**, in terms of more primitive operations.

Design: Data Abstraction

- **Data abstraction** is a methodology that enables us to isolate how a compound object is **used** from the details of how it is **implemented**, from more primitive data objects.

Design: Correctness Proof

- **A Program and its proof should go hand-in-hand,** with proof usually leading the way.
  - "Program testing can be a very effective way to show the **presence** of bugs but it is hopelessly inadequate for showing their **absence.**"
  - Example: Proof of termination of loops.

Coding

Coding

- “The tools we are trying to use and the language or notation we are trying to express or record our thoughts are the major factors determining what we can think or express at all.”
  - “Program **into** a programming language, not **in** it.”

Coding

- “A cluttered programming language can hinder us from thinking clearly; a restricted language can hide the best algorithm from us.”
  - **Cluttered Language:** Use of Roman numerals for large numbers
  - **Restricted Language:** Absence of recursion

Testing

Nil

Maintenance

Nil

Conclusion

- “It is a severe mistake to think that the programmer’s product is the program he writes; the programmer has to produce and present it with convincing arguments.”

Product = Specification + Program + Proof

References

- Dijkstra E. W. **On the role of scientific thought**, EWD447. http://www.cs.utexas.edu/users/EWD/.
- Dijkstra E. W. **The humble programmer**, ACM Turing Award Lecture, EWD340. http://www.cs.utexas.edu/users/EWD/.
- http://www.cs.utexas.edu/ EWD/

THANK YOU

Your comments are welcome.

Comments may be sent to the email id:

vpaleri@nitc.ac.in