

CS 342: Object-Oriented Software Development Lab
Problem Solving and Reporting

David L. Levine
Christopher D. Gill
Department of Computer Science
Washington University, St. Louis
levine,cdgill@cs.wustl.edu

<http://classes.cec.wustl.edu/~cs342/>

Problem Solving Approaches

- Why Use the Scientific Method?
- What is the Scientific Method?
- Solving Programming Problems
- Design Problems
- Debugging Problems
- “Problems”

1

Why Use the Scientific Method?

- Scientists have been *solving problems* for a long time
- Computer science is a branch of science
- Program design and implementation involves *problem solving*
- Therefore, programmers should use scientific approaches!
- And, they work!

2

What is The Scientific Method?

The scientific method is the process by which scientists, collectively and over time, endeavor to construct an accurate (that is, reliable, consistent and non-arbitrary) representation of the world.

An accurate representation is required for understanding!

3

Phases of the Scientific Method

1. Observation and description of a phenomenon or group of phenomena.
2. Formulation of a hypothesis to explain the phenomena. In physics, the hypothesis often takes the form of a causal mechanism or a mathematical relation.
3. Use of the hypothesis to predict the existence of other phenomena, or to predict quantitatively the results of new observations.
4. Performance of experimental tests of the predictions by several independent experimenters and properly performed experiments.
5. Iteration to demonstrate the validity of the hypothesis over time, and refine it if necessary.

Frank Wolfs, http://teacher.nsr1.rochester.edu/phy_labs/AppendixE/AppendixE.html

4

Solving Programming Problems

- Many design, implementation, debugging, *etc.*, problems can be phrased as testable hypotheses.
- Structured approach guides designer or programmers towards a solution.
- In addition to solving the problem at hand, structuring the solution for *reuse* reduces cost and development time for future products.

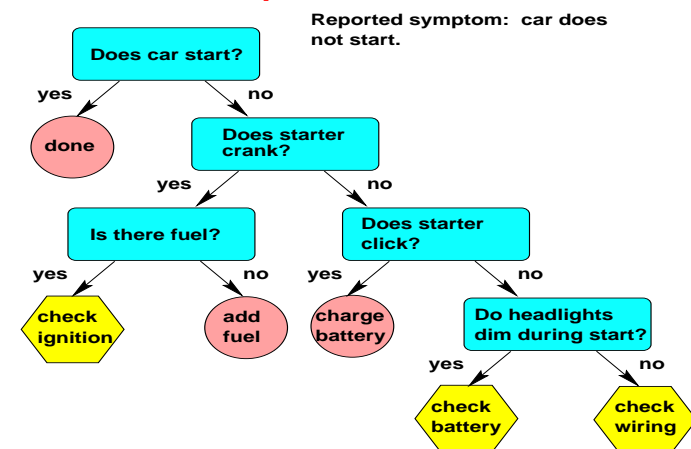
5

Structured Approach to Solving Design Problems

- *Observe and describe* solutions to previous design problems, relying on knowledge of domain experts.
- *Hypothesize, i.e.*, develop a *design pattern* that explains these solutions.
- *Predict* the result of application of the pattern to a new problem (observation).
- *Test* the application of the pattern via implementation of the design.
- *Iterate*, to refine as necessary.

6

Example: Decision Tree



7

Suggestions for Attacking Problems

- View problems as challenges, not impediments to progress
- Separate accidental from inherent complexity
 - Invest time in solving inherent complexities
 - Follow structured approaches
 - Quickly isolate and solve or resolve accidental complexities
- Isolate, hypothesize, predict, test, repeat

8

Problem Reporting

- Why Study Problem Reports?
- Example Problem Report Forms
- Effective Problem Reporting

9

Why Study Problem Reports?

- Problem reports contain minimum amount of information that must be conveyed to others
- Problem report forms abstract and categorize this information
 - Abstraction helps guide initial analysis for new problems: What do I look for?
 - Categorization helps organize the problem analysis
 - Good problem report forms are developed by domain experts
- Hmm, does this sound like a pattern?

10

Compiler Problem Report Form, page 1

General Information

Date	:	Version	:
Name	:	Release Date	:
Company	:	Host Machine	:
Phone	:	Host OS	:
FAX	:	Target CPU	:
Email	:	Target OS	:
Reference ID	:	(Your internal tracking number, if any)	
Report Type:		Problem or Error	()
		Enhancement Suggestion	()
		Documentation	()
		Other	()

11

Compiler Problem Report Form, page 2

Phase and Impact

Product(s): Builder/Driver ()
 Compiler/Assembler ()
 Linker/Object utilities ()
 Libraries/Header files ()
 Simulator/Monitor ()
 Debugger ()
 Editor/Version Control ()
 Other ()

Customer Impact: Severe Impact ()
 Moderate Impact ()
 Minor Impact ()
 No Significant Impact ()

12

Compiler Problem Report Form, page 3

General Problem Information

Problem Type: Tools hang, abort, or crash ()
 Fatal error message ()
 Unexpected warning message ()
 Incorrect behavior ()
 Inefficient/imperfect behavior ()
 Undetected error condition ()

Program Type: Test suite ()
 Benchmark ()
 System Software ()
 Customer Application ()
 Sample Test Program ()
 Incorrect Program ()

13

Compiler Problem Report Form, page 4

Specific Problem Information

Problem Summary (2 lines);

Command Sequence to Demonstrate Problem:

Commands shown in ~Progress window or with -v option in driver:

Header comment from .s .lst and/or .map files:

Minimal Test Program (Maximum 20 lines):

Exact Error Message, Incorrect Output or Behavior:

Desired Output or Behavior:

Workaround (if any):

14

The ACE Problem Report Form

To: ace-bugs@cs.wustl.edu
Subject: [area]: [synopsis]

ACE VERSION: 4.6.12

HOST MACHINE and OPERATING SYSTEM:
TARGET MACHINE and OPERATING SYSTEM, if different from HOST:
COMPILER NAME AND VERSION (AND PATCHLEVEL):

AREA/CLASS/EXAMPLE AFFECTED:
[What example failed? What module failed to compile?]

DOES THE PROBLEM AFFECT:

COMPILATION?

If so, what do your \$ACE_ROOT/ace/config.h and

\$ACE_ROOT/include/makeinclude/platform_macros.GNU contain?

LINKING?

15

On Unix systems, did you run make realclean first?
EXECUTION?
OTHER (please specify)?
[Please indicate whether ACE, your application, or both are affected.]

SYNOPSIS:
[Brief description of the problem]

DESCRIPTION:
[Detailed description of problem. Don't just say "<blah>
doesn't work, here's a fix," explain what your program does
to get to the <blah> state.]

REPEAT BY:
[What you did to get the error; include test program or session
transcript if at all possible.]

SAMPLE FIX/WORKAROUND:

Effective Problem Reporting

- Problem reports must be **specific**. "It doesn't work" is the antithesis of a specific problem report.
- Problem reports must be **reproducible**. If someone else can't find the problem, it's unlikely they'll be able to help. Also, documenting the problem so that other can reproduce it often provides **you** with more insight.
- For these and other good bug report writing tips, see <http://www.mozilla.org/quality>