



Testing the Readability and Composability of  
Computer Programs: Methodology

Kevin C. McCoy  
Department of Computer Science  
University of Oregon  
Eugene, Oregon

### Introduction

Recently I was pleased to see a paper by Weissman (1) on experimental methodologies for analysing the factors affecting program complexity and comprehensibility. The elucidation and analysis of these factors and corresponding principles of programming should be of interest to the programmer, language designer, and educator. Since this type of research is so fraught with methodological pitfalls I will add my recent experience hoping that it may aid others.

I have limited my investigations to the effects of control structure on the composition and comprehension of computer programs. I have done this with the aid of FLECS (2) a FORTRAN preprocessor, which adds control structure to FORTRAN without changing or extending its capabilities. In this manner we may concentrate our attention on a single factor while reducing emphasis on other areas such as documentation and data structures. In our first experiments we used programmers who already knew FORTRAN and had experience with FLECS as subjects. The selection of FORTRAN and FLECS groups was random.

### Composition

The complete description of a computation, including sample input and output, is given to both groups of subjects. They were allowed ten minutes to study the description at which time they may begin coding, one group in FLECS, the other in FORTRAN. After 50 minutes the subjects are told to stop. Copies are made of their programs. They are then instructed to debug their programs. The subject is asked to make the minimum number of changes necessary to obtain a correct program. All debugging runs are handed in with changes circled on the output listing, each change being classified into one of four areas: spelling, syntax, control, and compute.

Measures used to compare the two groups are the number of debugging runs and the number of control structure changes.

### Comprehension

The FLECS and FORTRAN group are given a description of a computation and a program which performs it. The subjects were allowed to study the description for ten minutes and may then proceed to fill in the missing parts of the program. The programs had every tenth token deleted; where a token is an identifier, symbol or number. The measure used to compare the groups was the percent correct answers.

One other method was tried: matching descriptions of a computation with code segments. Here the creation of equivalent tests was difficult.

### Discussion

For testing readability and comprehensibility, the selection of test programs is the most difficult task. In cases where only limited times are available the problems which are manageable are often trivial. It would seem that programs should be selected that have a moderate degree of complexity and that the duration of the experiment should be increased, rather than limiting complexity to fit a time limit.

For testing composition the creation or selection of subjects is a difficult task. The ability and ease with which a programmer approaches a task is governed by the control which he has over his instrument. Clearly subjects should have a complete and thorough understanding of the language and programming methods being tested. This problem we are attempting to solve by the use of parallel classes, one using FORTRAN and the other FLECS.

- (1) Weissman, L. "Psychological Complexity of Computer Programs: An Experimental Methodology", SIGPLAN Notices, vol. 9, no. 6, 1974.
- (2) FLECS is a FORTRAN preprocessor intended to add control structure to FORTRAN. It was developed by W. Terry Beyer.