# Research Journal

# COMP110

# 1608289

Paper 1:

'Experimental Investigations of the Utility of Detailed Flowcharts in Programming'

Notes:

This paper is about the preliminary flowcharts when producing or debugging a program. The paper explains a few different experiments that were undertaken to test whether the use of designing a program with the aid of flowchart will aid the user in creating it to decrease the possibility of bugs throughout the code which will cause the program to be unable to run as intended. The experiments took place from a couple of universities: Indiana University and also Purdue University.

The first experiment was based on composition. The method of the experiment is as follows: Thirty-four students, received test instructions which indicated that they were to write a flowchart and then program for a given problem. The flowchart counted for 15 points, the program 25 points. Twenty-eight subjects were instructed to merely write the program, which counted for full 40 points. The subjects were given as much time as they wanted to complete the test. The grading was done by a graduate student with much experience in grading programs and in consulting students with programs. [1]

The results for the experiment were very close and also surprising. Having a flowchart to plan a program before creating it seems like a more soundproof idea than simply programming it without any form of direction. The results are as follows: The flowchart group mean score was 94 while the nonflowchart group mean was 95. [1] Although the results show no significant difference the two variants of the experiment, it shows that those who didn't use a flowchart to plan their program had achieved a higher grade than those who did.

Paper 2:

Letters to the Editor: Abbreviations for Computer and Memory Sizes

Notes:

In this paper, the author, Donald R. Morrison, explains his thoughts on the abbreviation for computer and memory sizes. Throughout the short article he emphasises that when doubling the size of a 32K results in receiving a 65K. This is due to the rounding of the sizes of a 32K to being actually 32,768. The confusion arises because we use K, which traditionally means 1000, as an approximation for 1024. [2] The author carries on by stating that as memory sizes get larger and eventually into the millions that users could use $32k^2$ and also $64k^2$, this is so the users will not need to explain the discrepancies between what they said and what they meant. [2]

Paper 3:

Language Protection by Trademark Ill-advised

Notes:

In this paper, the author, Bernard A. Galler, explains his disgust within the licensing put in place by Rockford Research for all forms of Trac languages. The creator of the Trac languages, Calvin N. Mooers, believes that “well-drawn standards are not enough to prevent irresponsible deviations in computer languages”. [3] Calvin N. Mooers believes that it shouldn’t just be for commercial uses but in fact for academic institutions too. Bernard A. Galler then carries on his argument that it shouldn’t be licensed by addressing the creators of Snobol, Bell Laboratories. He compares the use of Comit, another language which was licensed, Bell Laboratories created Snobol and released it on a commercial and academic use for all. This latter language and its software inevitably superior [3] compared other languages such as Comit which required users to pay a fee to be able to make use of. The primary reasoning behind the licensing of Trac have actually proven to counteract the problem that Calvin N. Mooers had tried to prevent, that the software would allow “irresponsible young people” [3] to create malicious extensions for the language. Snobol had allowed the users to create extensions which had furthered the coherence and popularity of Snobol whereas Comit enjoys relative obscurity. [3] It would seem that Bernard A. Galler was trying to prove a valid point to Calvin N. Mooers about removing the licensing of his language of Trac.

Paper 4:

A Fast Procedure for Computing the Distance Between Complex Objects in Three-Dimensional Space

Notes:

In this paper the author explains the importance of the distance between two or multiple three-dimensional objects and how to compute the distance between the two or more. I should find more time to research the method of computing the distance between two three-dimensional objects by reading this paper in much more depth.

**References**

1. Experimental Investigations of the Utility of Detailed Flowcharts in Programming
2. Letters to the Editor: Abbreviations for Computer and Memory Sizes
3. Language Protection by Trademark Ill-advised
4. A Fast Procedure for Computing the Distance Between Complex Objects in Three-Dimensional Space