Student Mistakes in Introductory Programming: Sample Problems

Amitrajit Sarkar, Michael Lance, Mike Lopez, Robert Oliver, Luofeng Xu

Christchurch Polytechnic Institute of Technology
130 Madras Street,
Christchurch, New Zealand
+64 3 940 8000

sarkara@cpit.ac.nz, lancem@cpit.ac.nz, lopezm@cpit.ac.nz, oliverr@cpit.ac.nz, xul@cpit.ac.nz

ABSTRACT

Learning to program is a challenging task for novice learners. This study aimed to investigate students' concepts as they were being formed. To capture these, we chose to focus on students who made some mistakes in basic concepts. Our study sought to capture students' conceptions at a very early stage in their study: five weeks into an introductory programming course. We invited students who did not pass an early mastery test at their first attempt to participate in a diagnostic and remedial session. In this poster we look at a sample of the original problems given to these students and an analysis of the number of questions incorrectly answered by each of the students that failed the early mastery test.

Categories and Subject Descriptors

K.3.2 [**Computer and Information Science Education**]: Computer Science Education – *programming misconceptions and mistakes.*

General Terms

Human Factors.

Keywords

Misconceptions, alternative conceptions, novice programming, programming mistakes.

This poster paper appeared at the 4th annual conference of Computing and Information Technology Research and Education New Zealand (CITRENZ2013) incorporating the *26h Annual Conference of the National Advisory Committee on Computing Qualifications*, Hamilton, New Zealand, October 6-9, 2013. Mike Lopez and Michael Verhaart, (Eds).

1. INTRODUCTION

Learning to program is a challenging task for novice learners [2]. Beginning programmers need to master a wide range of concepts [1] and consequently, it is not surprising that many wide scale studies have found that novices are struggling at the early stage of this learning process.

The present research takes its starting point from a phenomenographic study on novice students' understanding of the concepts object and class [2]. This poster gives a visual perception and a quantatitive view of the results gathered so far. We believe that it is important to recognise the types of mistakes that students make at an early stage of their learning so that remedial action can be taken as soon as possible to allow students to develop robust program code and apply "best practice" to the design of the solutions to problems.

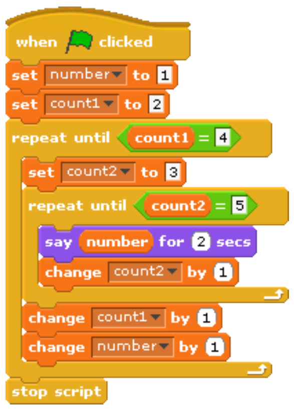
2. METHOD

In the first five weeks of the course, the Scratch programming language is used in tutorial sessions to introduce fundamental concepts that underpin programming: variables, sequence, selection, repetition, recursion, and message passing. At the end of this segment, students take a test which assesses their ability to trace and analyse code. The test has six questions; for each question, the student has to indicate the correct output of a short program. Students are expected to carry out hand execution (by way of formalised desk checks taught during the first five weeks of the course) of the code to determine this output.

We approached the students who did not pass the test at their first attempt and invited them to participate in a diagnostic and remedial session. All students chose to participate in this session. Members of the teaching team were assigned to students and worked with them on a one-to-one basis. The assigned teaching team member reviewed the student's test answers with the student and asked the student to demonstrate how they carried out the desk-checking tasks that gave rise to the answer they had submitted.

One of the six questions is shown below:

What is the output of the code?



```
when ⚑ clicked
set number to 1
set count1 to 2
repeat until count1 = 4
    set count2 to 3
    repeat until count2 = 5
        say number for 2 secs
        change count2 by 1
    change count1 by 1
    change number by 1
stop script
```

Choose one answer.

a. 1 1 2 2

b. 1 2 2

c. 1 2

[✓ d. 1 1 2 2 3 3]

Figure 1: Example Question

3. FINDINGS

In this section we present an analysis of all six students' results from the original test and an analysis of the results of the four students who sat the resit test. (C = Correct and I = Incorrect).

Table 1: Student answers in the original test.

Student	Q1	Q2	Q3	Q4	Q5	Q6
A	I	I	C	C	I	I
B	I	I	C	C	I	I
C	I	I	C	C	I	I
D	I	I	C	C	I	I
E	I	I	C	C	I	I
F	I	I	I	C	I	C

Table 2: Student answers in the resit test.

Student	Q1	Q2	Q3	Q4	Q5	Q6
A	C	C	I	C	C	C
B	C	C	C	C	C	C
C	C	C	I	C	C	C
D	C	C	C	C	C	C

4. DISCUSSION

How a student attributes the reason for lack of success is important because it affects the student's motivation to remedy the lack of success by engaging in further learning. For example, an attribution to luck which is external and uncontrollable would create little motivation to learn. Similarly, an attribution to internal stable factors such as personality would create little motivation. However, all of the students attributed the reason for their lack of success to an internal locus of control, believed that the outcome could be modified, and believed that this was under their control. This combination suggests that the application of remedial techniques could be effective and supports the approach of offering students who did not succeed on their first attempt the option of attending a diagnostic session with subsequent remedial instruction.

A student is also affected by their motivation for the subject. For example one of the students has a previous degree outside the computing field and was doing the course to develop their understanding of computing rather than to gain accreditation. They has not yet decided on further study. A second student subsequently chose a study path leading to a major in Information Systems, while a third student chose a study path leading to a major in Software Engineering.

5. CONCLUSION

We found that the technique of inviting students who do not succeed in a test to participate in an in-depth diagnostic interview and one-on-one remedial instruction was useful, even though no major misconceptions were identified. Indeed, we found that the lack of success in the test was attributable more to application of process than to conceptual misunderstandings or alternative conceptions. However, our diagnostic interviews focused closely on performance in the test and it is possible that a more broadly focused interview would have discovered alternative conceptions.

We plan to continue to explore and refine the use of this technique. In particular, we will use a broader approach in our diagnostic interviews to attempt to elicit alternative conceptions.

6. REFERENCES

1 du Boulay, D. *Some difficulties of learning to program. In Studying the Novice Programmer*. Lawrence Erlbaum, NJ, 1989.

2 Eckerdal, A, Laakso, M, Lopez, M, and Sarkar, A. Relationship between text and action conceptions of programming: a phenomenographic and quantitative perspective. In *Proceedings of the 16th annual joint conference on Innovation and technology in computer science education (ITiCSE '11)* (New York 2011), ACM, 33-37.