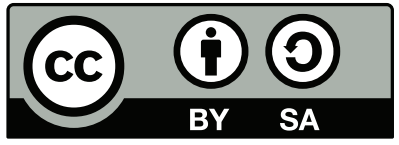
Laboratory of Computer Science Education

Misconceptions

Alberto Montresor

Università di Trento

2020/04/07



Motivation

Raising awareness of students misconceptions and other difficulties, and strategies for addressing them, can help computer science teachers to better teach their students [Quian and Lehman, 2017]

Misconceptions are consistent, systematic, are based on some modification of correct knowledge, and are stable in attempts to change them.
Teachers should look for misconceptions and help learners correct them by addressing their source.

Misconceptions

Misconceptions

Sorva defined misconceptions as understandings that are deficient or inadequate for many practical programming contexts [Sorva 2013]

Other names

- Alternative conceptions (less negative)
- Difficulties, errors, bugs, mistakes
- Naive conceptions/knowledge, intuitive knowledge

Difficulties in Strategic Knowledge

Examples

Parenthesis, semicolons, operators, ...

We do not discuss them in depth here, because problems in syntactic knowledge are often easy to detect and fix. Perhaps, that is why they are often noted as the most frequent mistakes novices make

Personalmente: la differenza fra disattenzione e mancata conoscenza del linguaggio

Difficulties in Conceptual Knowledge

- Students may fail to understand that variables can only hold one value at a time
- The order of statements assigning values to variables is important
- Students may also mistakenly believe that the English meaning of the variable name affects the value of the variable
- students often used local variables and global variables inappropriately. statements in both if and else blocks of a conditional expression will be executed
- Students even mistakenly think that if the condition of an if-statement is false the execution of the whole program stops
- Novices often believe that one looping construct is better than the other one

Difficulties in Conceptual Knowledge

- Students may mistakenly believe that a conditional statement inside a loop will be executed whenever the condition is true, even if this occurs outside of the loop
- Even understanding a three-line swap is difficult for novices
- Misunderstandings about references and reference assignment are also common among students in introductory programming
- When a novice programmer debugs a program, he or she typically reads and traces code in a local manner line by line without a holistic view about programming

Difficulties in Strategic Knowledge

Students in introductory programming may know how to write a runnable program but fail to check the appropriate boundaries of conditions and unexpected cases in their program

Another strategic challenge for students is to merge blocks of code that should be applied together. For example, when solving a problem requires interleaving two or more patterns, novices often only concatenate the patterns and bypass the interlaving.

Factors contributing to misconceptions

- Task Complexity and Cognitive Load
- Natural Language (Natural way of thinking)
- Math Knowledge
- Flawed Mental Models
- Inadequate Patterns and Strategies
- Environmental Factors
- Teachers Instruction and Knowledge

Cognitive load

In a study of CS1 students, Sanders and Thomas [2007] found that most of the students submitted flawless programs for the first assignment but when tasks became more challenging students started to make more syntactic mistakes

By analyzing student errors in programming, Anderson and Jeffries [1985] found that the complexity of the task made students unable to retain information in their working memory and led them to omit necessary parts of the code.

Teacher Instruction and Knowledge

Novice computer science educators, however, face difficulties in gaining the notion of alternative conceptions. Specifically, they face difficulties in

- Understanding how people do not understand topics which they conceive as trivial ones;
- Lowering their level of understanding to that of a novice learner since their understanding of the subject area is usually more advanced;
- "Getting into the head" of someone else (not just a pupil) because they have not gained yet the experience needed for performing such tasks.

Errors are good

Mistakes should not necessarily be conceived negatively since they provide learners with the opportunity to correct their current knowledge and update their mental representation of the said topic or concept.

One of the main messages that should be delivered to the teachers is that a learning opportunity exists in every pupil's mistake (or misunderstanding)

Reacting to errors

A common assumption held by novice computer science teachers is that pupils' questions should be answered immediately and their problem should be solved right away.

Therefore, many teachers tend to fix an error or a mistake in pupils' solutions or pupils' ways of thinking as soon as they recognize it.

Teachers, however, should resist this tendency and listen to their pupils very carefully before they start seeking for the corrective teaching action.

In practice, as it turns out, in many cases, when a pupil asks a question, it is worthwhile just repeating the pupil's question, allowing the pupil to answer it first.

Reacting to errors

A teacher can ask pupils a series of questions whose purpose is to reveal the pupils' alternative conceptions.

Metaphorically, the teacher can develop a series of tests to expose pupils' cognitive bug.

It is important to recognize the bug both from the pupil's and the teacher's perspectives; specifically, the pupil must recognize the bug in order to start debugging and modifying his or her cognitive model; the teacher should expose the pupil's bug in order to implement some pedagogical intervention.