Learnt Helplessness in Computing Education by Phil Bagge

Strategies to turn helplessness into resilient problem solving	Tick if you recognise this issue
Process is more important than outcome. If pupils are on task and puzzling something out give them time to do so. It is better to have 60% finished where that work is mainly the pupils own work than 100% finished where a teacher or pupil next to them solved it for them.	
It is very important to establish a positive class attitude towards problem solving. Computing science is very useful in that it calls errors bugs and finding errors debugging. Although all bugs are caused by humans, the language is much more impersonal than mistakes which imply blame or fault. Using bug and debugging language is helpful.	
It is also important to let pupils know that mistakes/bugs are a normal part of computing, that they are to be expected, that professional programmers write code that have bugs all the time and that the teacher will not be cross or upset if their work has bugs/mistakes.	
Promote the idea that it is not the teachers' job to fix their algorithms or debug their code. It is the teachers' job to promote useful strategies that they can use to fix things themselves. So when they come to a teacher they know they are looking for strategies to find and fix things themselves.	
For those pupils transitioning from learnt helplessness to useful problem solving they need to see what they are doing. Ask pupils; 'are you trying to get me to do your work for you' 'Are you trying to get me to solve the problem for you?' Some pupils won't be able to move on until they recognise their work avoidance strategy for what it is. Good teachers do this tactfully and with regards to pupils known issues, but an element of gentle challenge is inevitable to identify the issue.	
Encourage the class to join you in this by putting a ban on doing things for other people. They can describe what to do but are not allowed to do it for them or give them a full solution to programming solutions. As you model this, pupils will reflect your attitude to their peers. Have a ban on touching anyone else's mouse, keyboard or touchscreen. You can compare this to writing the answers in someone else's maths or literacy exercise book.	
Move pupils away from language that personifies digital machines. "My computer hates me," is typical. Miles Berry describes computers as deterministic which means that if all the inputs are the same you will always get the same output. Personification encourages pupils to think that an answer might not be available due to the capriciousness of the machine, an attitude that is anti-problem solving and incorrect[1].	
Don't neglect the other adults in the class, all your good work could be being undone by your LSA or classroom assistant. Train them to help using good strategies and hints rather than solutions. If you are providing training on computing don't neglect your class room assistants/LSAs, they are important.	

[1] It maybe that the problem is beyond your technical ability which is fine and why we have experts

You may notice learnt helplessness in teachers and other staff. It is worth the hassle to challenge this in staff training? As parents and teachers, we know that children don't do what we say but what we do. This is just as true in the classroom or computer suite. We need to be tactful but if we don't identify the problem, nothing will change.	