

Teacher Lesson 7: Plan, Test, and Debug a Program

As your students develop more complex projects and longer programs, they need to develop a good planning strategy. Help them learn to:

1. Define your project. Exactly what do you want Tuff-Bot to do?
2. Plan out the code before you touch Tuff-Bot. Write down the commands you think will work to complete your project successfully.
3. Give Tuff-Bot the commands you expect will work.
4. If Tuff-Bot successfully completes the project, great job! You planned well!
5. If Tuff-Bot did not do what you expected and didn't complete the project successfully, take a close look at what it did do. Where did things go wrong? You need to debug your program. Can you revise your code so that it will work the next time? Try your revised project and see if it works this time.

Do you know where the word “debug” came from?
[from https://en.wikipedia.org/wiki/Grace_Hopper]

While Grace Hopper was working on a Mark II Computer at Harvard University in 1947, her associates discovered a moth that was stuck in a relay and impeding the operation of the computer.

Upon extraction, the insect was affixed to a log sheet for that day with the notation, “First actual case of bug being found”. While neither she nor her crew members mentioned the exact phrase, “debugging”, in their log entries, the case is held as a historical instance of “debugging” a computer and Hopper is credited with popularizing the term in computing.

For many decades, the term “bug” for a malfunction had been in use in several fields before being applied to computers. The remains of the moth can be found taped into the group’s log book at the Smithsonian Institution’s National Museum of American History in Washington, D.C.



Share this fun story with your students!