Questioning for Learning

Overview:

Questioning has long been recognized as a great way to stimulate a dialogue, gain the perspectives of others and ultimately support learning.

These benefits only materialize when we ask the 'right sorts' questions. For instance a closed question such as;

"Do you understand this code?"

is a very different question (and elicits a very different response) from an open question such as;

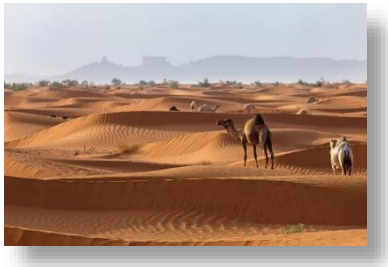
"tell me what you understand about this code"

To increase the potential for learning, the following offers a few questions for you to ask during your evaluations. These questions should be used along with the Project Evaluation Sheets, not as a replacement.

We are not expecting you to ask all the following questions, but certainly we are expecting you to ask a couple of questions from each of the following themes.

- Big Picture Questions
- Detailed Questions
- Reflecting on Learning Questions

Big Picture Questions



These questions are intended to see how well students know what '*problem the code is solving*', and in broad terms

Example big picture questions include:

- Without looking at the code (or talking specifically about the actual code), what is the purpose of the program / function?
- What was the logic you used to solve the problem?
- In pseudocode, tell me what is happening and how the code works
- Draw a flow chart of the code and its major operations
- How can this project be adapted to solve other problems?
- How is this project different from other projects?
- How is this project similar to other projects?

Detailed Questions

This questions allow you to 'focus on the code and seek to understand precisely what is happening'

Example detailed questions include:

- Looking at your code, describe what is going on in *'this function'.'*
- Tell me about a couple of scenarios. For instance, "*if the arguments of 'this function are equal to abc, what would happen? What happens if the arguments are now xyz?"*
- What test scenarios have you created and performed to be sure the code operates as expected?
- What errors / unexpected input etc., are you catching in your code and how are you catching them?
- How would you develop the code further to perform additional / new (related) functionality?
- Write a Main to show me how this function works.
- How else could you write *'this function*?'

Reflecting on Learning Questions

These questions focus the student's mind on what they have learned from working on the project, and how they will use that learning for future projects.

Example reflecting on learning questions include:

- What did you find most challenging about this project?
- What coding skills have you developed by completing this project?
- In relation to this project, what are you still a little uncertain about?
- What will you do now to strengthen the areas you are still uncertain about?
- What questions did you not want me to ask you?
- What questions would you ask a student that you are evaluating (on this project) to know they understand their code and see what they have learned?
- How will you document you learning so that you can use that learning in future projects / work?

What next…?

- Ask a couple of the questions from each theme during your evaluations; alongside the conventional evaluation)
- Provide a comment on the student's responses to your questions in the evaluation feedback (as an evaluator).
- Provide a comment on the evaluator's questioning strategy in the evaluation feedback (as a student).
- Contact us if you have other questions you think would make great additions to the above list.

Mark@42abudhabi.ae

Afaq@42abudhabi.ae