✓ **Congratulations! You passed!**

Grade received 100%   **To pass** 80% or higher

[**Go to next item**]

---

1. When should we respond to comments from collaborators and reviewers?   1 / 1 point

   ◯ When their comments address software-breaking bugs

   ◯ No need, just resolve the concerns and be done with it

   ⦿ Always

   ◯ Only when a code correction is necessary

   ✓ **Correct**
   Excellent! It is good manners and proper conduct to respond, even when it's simply an acknowledgement.

2. What is a nit?   1 / 1 point

   ⦿ A trivial comment or suggestion

   ◯ A couple lines of code

   ◯ A repository that is no longer maintained

   ◯ An orphaned branch

   ✓ **Correct**
   Good work! In git jargon (and elsewhere in the tech world), a nit is a minor "nitpick" about a piece of code.

3. Select common code issues that might be addressed in a code review. (Check all that apply)   1 / 1 point

   ☑ Using unclear names

   ✓ **Correct**
   Excellent! Unclear names can make our code hard to understand.

   ☐ Following PEP8 guidelines

   ☑ Forgetting to handle a specific condition

   ✓ **Correct**
   Alright! If there is a specific condition that could cause a problem and we don't address it, the result could be catastrophic.

   ☑ Forgetting to add tests

   ✓ **Correct**
   Woohoo! Tests are an important addition to our code to ensure it runs smoothly.

4. If we've pushed a new version since we've made a recent change, what might our comment be flagged as?   1 / 1 point

   ◯ Accepted

   ◯ Resolved

   ⦿ Outdated

   ◯ Merged

   ✓ **Correct**
   Nice job! If we push a new version after making a change, old comments are marked with the "Outdated" flag.

5. What are the goals of code review? (Check all that apply)

1 / 1 point

☑ Make sure that the contents are easy to understand

> ✓ **Correct**
> Right on! By reviewing our code, we can identify where we can make our code more clear and easy to understand.

☑ Ensure consistent style

> ✓ **Correct**
> Awesome! By comparing our code to style guidelines, we can keep our style consistent and readable.

☐ Build perfect code

☑ Ensure we don't forget any important cases

> ✓ **Correct**
> Good job. Code review can reveal cases or conditions we need to handle in our code.