## Congratulations! You passed!

Grade received 100% To pass 85% or higher

Go to next item

# Programming Assignment Quiz (Do programming assignment FIRST)

**Latest Submission Grade 100%**

**1.** Does your program correctly respond to mouse clicks and mouse hovers, as described in the project assignment write up? This should include the following behaviors (test them now): 1 / 1 point

- When I city or earthquake is hovered over, information should be displayed about that city or earthquake. Test this by hovering over several cities and earthquakes now.
- It should never be the case that information for more than one item is displayed at the same time. Test this last point explicitly by hovering over two nearby cities or earthquakes, and a city next to an earthquake.
- If all earthquakes and cities are displayed, when you click on an earthquake, all other earthquakes should be hidden and all cities except those in the threat circle should be hidden. Test this by clicking on an earthquake now.
- If earthquakes or cities are hidden, clicking anywhere should re-display all earthquake and city markers. Test this by clicking somewhere now. Try to test clicking on where a hidden earthquake is to make sure it's not selected.
- If all earthquakes and cities are displayed, when you click on a city, all other cities should be hidden and all earthquakes except those that affect the city (city is in their threat circle) should be hidden. Test this by clicking on a city that has some nearby earthquakes now.
- If you click on no marker and everything is already displayed, nothing should happen. Test this now.

- (•) Yes
- ( ) Not quite...

**Correct**
Congratulations! You have created an interactive program.

**2.** Select the classes that are the *direct subclasses* of (i.e. directly extend) the class CommonMarker in module 5 programming assignment (when the assignment is complete). Select ALL that apply. 1 / 1 point

- [ ] LandQuakeMarker
- [x] CityMarker

**Correct**
Yes, CityMarker extends CommonMarker. You can see this in the class declaration header.

- [ ] OceanQuakeMarker
- [ ] SimplePointMarker
- [x] EarthquakeMarker

**Correct**
Yes, EarthquakeMarker extends CommonMarker. You can see this in the class declaration header.

**3.** What caused the bug in the starter code when you modified CityMarker to extend CommonMarker instead of SimplePointMarker? 1 / 1 point

- (•) CityMarker did not implement the drawMarker method, which is abstract in CommonMarker.
- ( ) The header in CityMarker needed to say "implements CommonMarker" instead of "extends CommonMarker"
- ( ) CommonMarker cannot be extended because it is an abstract class.
- ( ) The body of the showTitle method in CityMarker was empty

**Correct**
This is the correct response. All abstract methods must be implemented by a non-abstract class that extends an abstract class. The solution was to change the draw method in CityMarker to be named drawMarker, which was then called from draw in CommonMarker.

**4.** What common functionality does CommonMarker implement for its subclasses? In other words, what functionality is inherited by subclasses (not overrided) and used 1 / 1 point

either by users of subclass objects or by the subclass itself? Select ALL that apply.

- [x] The setClicked method that allows the user to set the "clicked" state of the marker.

**Correct**
The setClicked method sets the "clicked" state of the marker. None of the subclasses override this method.

- [x] The draw method

**Correct**
The draw method is implemented in CommonMarker and is not implemented in any of its subclasses. drawMarker is the method that the subclasses implement.

- [ ] The showTitle method that displays the marker's text
- [ ] The drawMarker method that draws the details of the marker

**5.** Where are the variables mouseX and mouseY, which you used in the method selectMarkerIfHover, declared? **1 / 1 point**

- ( ) They are local variables declared in selectMarkerIfHover
- (•) They are member variables of the PApplet class and are inherited by EarthquakeCityMap
- ( ) They are parameters passed to selectMarkerIfHover
- ( ) They are member variables explicitly declared in EarthquakeCityMap.

**Correct**
This is the correct response. They are PApplet member variables and so they can be used in EarthquakeCityMap because it extends PApplet.

**6.** In which class(es) is the code that *actually draws the text label* (i.e. the call to the PGraphics text(...) method) next to the city markers and earthquake markers located? Select ALL that apply. **1 / 1 point**

- [ ] LandQuakeMarker
- [ ] EarthquakeCityMap
- [ ] OceanQuakeMarker
- [x] EarthquakeMarker

**Correct**
This class is one of the classes that draws the text next to the marker in its drawMarker method.

- [x] CityMarker

**Correct**
This class is one of the classes that draws the text next to the marker in its drawMarker method.

- [ ] CommonMarker

**7.** Consider the following code, which might be used as a helper method in the mouseClicked method for this programming assignment. This method is supposed to do the following: **1 / 1 point**

- If lastClicked is not null, it should abort the method.
- Otherwise it should check all of the cities for a click.
- If it finds a city that has been clicked and that city is not hidden and that city is the first one it finds to be clicked, it sets lastClicked to be that city's marker
- It then sets the rest of the cities to be hidden.
- If there is no city clicked, it does nothing to the city markers.

```
1  private void checkCitiesForClick()
2  {
3      if (lastClicked != null)
4          return;
5      for (Marker marker : cityMarkers)
6      {
7          if (!marker.isHidden() &&
8              marker.isInside(map, mouseX, mouseY) &&
9              lastClicked == null)
10         {
11             lastClicked = (CommonMarker)marker;
12         }
13         else {
14             marker.setHidden(true);
15         }
16     }
17
```

```
18    }
```

Does this code have a bug, and if so, what is it?

- ○ This code can potentially select more than one city, and making it so that more than one city remains unhidden.
- ◉ If no city is clicked, this code will hide all the cities (instead of leaving them alone like it is supposed to).
- ○ This code can potentially select a hidden city
- ○ It will run the loop to check the cities for a click even if lastClicked is not equal to null.
- ○ There is no bug. The code works correctly as described.

**Correct**
If no city is clicked, the "else" statement in the "for" loop will execute for every city, in effect hiding all of the cities.

**8.** In precise English, describe how your code handles and responds to mouse clicks. You should not include code in your description. You should include a short description (a couple sentences) for what each method does, as well as the overall flow of how the methods work together. **1 / 1 point**

Being able to describe technical code using English in a way a technical person who is unfamiliar with your code can understand is an important skill. So while this is an ungraded question, we encourage you to try it. We've included our explanation (of our code) so you can have a reference for what we are looking for after you attempt your own explanation. We'll ask you do to similar explanations in the peer review assignment at the end of module 6 as well as in future courses.

my code worked fine.

**Correct**
Here's how our code handled mouse clicks:

When the user clicks the mouse, the mouseClicked code in EarthquakeCityMap is called by Java. This method first checks the lastClicked variable. If it is null, meaning a city is already shown as "clicked", it sets lastClicked to null and unhides all the cities and earthquakes.

Otherwise, it relies on two helper methods: checkEarthquakesForClick and checkCitiesForClick.

checkEarthquakesForClick first checks lastClicked, and aborts if it is not null (just in case).
Then it loops through all the earthquakes to see if one has been clicked
on. If it finds one, it loops through all of the earthquake markers and sets
all but the clicked earthquake to hidden. Then it loops through the city markers and sets all of the city markers outside of the clicked earthquake's threat circle to be hidden. It then returns so that it does not check anymore earthquakes.

checkCitiesForClick first checks lastClicked, and aborts if it is not null (which could mean an earthquake has already been found as clicked). Then it loops through all the cities to see if one has been clicked on. If it finds one, it loops through all of the city markers and sets all but the clicked city to hidden. Then it loops through the earthquake markers and sets all of the earthquake markers for which the city is outside of the threat circle to be hidden. It then returns so that it does not check anymore cities.

**9.** How long, total, did you spend on this programming assignment, to the nearest hour? Include only the time you were actively working on the programming assignment including time you spent watching support videos or re-watching videos specifically because you needed help on the assignment. **1 / 1 point**

10hrs

**Correct**
Thank you for your response.