

CSC 20: Project 3 - Map

We have spent time in class learning about Collections classes and generic classes. This project is intended to solidify your understanding of Maps and generics.

Specification:

Write a class called `Map20` that is generic for both keys and values (ie, your class header should start `public class Map20<K,V>`). Normally a Map implementation would also advertise that it implements the Map interface, but we are going to make a reduced version of Map and so don't want to promise the whole interface.

Your class should implement the following public methods (you may implement any number of private methods if you want): no-argument constructor, `clear`, `containsKey`, `containsValue`, `equals`, `get`, `put`, `keySet`, `remove` (both versions), `replace` (both versions), `size`, `isEmpty`.

Your implementation must use the exact same header and behave exactly as described in the javadoc for Map, except for two exceptions. Your Map20 should not support null keys or values, so any `NullPointerException` documented in the javadoc should be implemented by you. No other exceptions need to be implemented. Your `keySet` method should not return a set "backed by the map" (instead it should return a normal set that contains all the same references in your Map's keys but is otherwise unconnected).

To Receive Credit:

Follow the directions in [Project Requirements](#) and [DBInbox Submission](#), and submit by 11:59pm, Sunday, April 2, 2017.

Questions?

If something is not clear, ask questions in class, on Piazza, or in office hours. Do not wait until the last minute to clear things up. Start early!

First published March 9 , 2017.

Modified March 13 , 2017: Changed from `<V,K>` to `<K,V>` and added no-arg constructor.

Modified March 24 , 2017: `keySet` returns independent set, null not allowed as key or value, `put`.