

Programming Assignment 5

Due at the beginning of your discussion session on
October 3-7, 2016

Reading

- Read the Quick Reference on routine names (blackboard).
- Read Sections 7 (preamble), 7.1, 7.2, 7.3, 7.5, 7.6, and 19.6 in Code Complete.

Grading Guidelines

An automatic C (or less) is triggered by:

- Any routine with complexity greater than 4, or by
- Any piece of code that is essentially repeated.



Programming

First, make all the changes discussed in your discussion section. Additionally, you should refactor your code to make sure that it adopts the principles covered in the reading assignments.

In this assignment, you will implement methods that circulate messages according to the device cabling.

Modify the Connector so that additionally `setPeer` throws an appropriate `ConnectionException` if the peer is already busy, and otherwise sets the peer's peer to this connector.

Modify the GoAmateur device so that it responds to binary messages by sending on all of its connectors the binary message 293.

Modify the hub so that it forwards any received message on all its connectors except the one from which the message was received.

In your tests, create a USB system containing at least two hubs, three printers (not all of the same type), and a Webcam. These USB devices should all be connected with each other, directly or indirectly. Then, explore scenarios in which:

- A string message is broadcast from a hub.
- A binary message is sent from a hub along a connector that links the hub to a Webcam.
- A binary message is broadcast from a hub.

For extra credit, visualize the USB system and the message traffic, for example with JavaFX or with GWT.

General Considerations

This assignment concludes the USB project. In the next assignment, we will turn to a different project.

These classes may contain as many auxiliary private methods as you see fit, and additional helper classes may be defined.

You should write JUnit tests to make sure that your primary methods work as intended. However, we will revisit testing later on in the course, so extensive testing is not yet recommended. Similarly, your code should have a reasonable number of comments, but documentation is going to be the topic of a future assignment. As a general guideline at this stage of the course, comments and tests should be similar to those accepted in EECS 132.

Discussion Guidelines

The discussion will focus on routines, including but not limited to appropriate routine names.

Submission

Bring a copy to discussion to display on a projector. Additionally, submit an electronic copy of your program to Blackboard. In addition to your code, include a README file explaining how to compile and run the code. The code should be handed in a zip, tar.bz2, or tar.gz archive. Archives in 7z cannot be accepted.