

# Assignment 1

- create a program for copying files
  - source and target files are given as program parameters
  - try copying using streams and using the NIO package (the methods `transferFrom/transferTo`) and also using `Files.copy()`
- change the previous program so that the second parameter is either file or directory
  - in the case of a directory, the file is copied to the directory and its name is not changed

# Assignment 2

- create the **adduser** program, which add a user to a UNIX system
  - users are in the file `/etc/passwd`
    - create a copy of the file and make changes to the copy
    - format of the file available in the manual page
      - `man 5 passwd`
  - program will be interactive
    - asks the user about his/her user-name
    - tests whether the user-name exists
    - suggests a UID
      - testes whether UID can be used
    - offers the home directory
      - `/home/username`
    - offers a shell

# Assignment 3

- update the “formatting” program from the previous practicals so it can take another optional parameter with a name of file, to which the result should be written
  - without it, it is written to the std output

Tests...

# Test 1

- What does the following program?

```
public class TestString {  
    public static void main(String[] args) {  
        String s = new String("Hello world");  
        System.out.println(s);  
    }  
}
```

```
class String {  
    private final java.lang.String s;  
    public String(java.lang.String s) {  
        this.s = s;  
    }  
    public java.lang.String toString() {  
        return s;  
    }  
}
```

- A cannot be compiled
- B prints Hello world
- C something else happens

# Test 2

- Is it possible to declare the class B, so that the program prints false? But without overriding the method equals!

```
public class A {  
    public static void main(String[] args) {  
        B b = new B();  
        System.out.println(b.equals(b));  
    }  
}
```

# Test 2

- Solution
  - overload the method equals
  - i.e. define the method

```
public boolean equals(B b) {  
    return false;  
}
```

- Continuation – and without overloading?

# Test 2

- Also yes

```
class B {  
    public B() {  
        System.out.println(false);  
        System.exit(0);  
    }  
}
```





Slides version PJ07.en.2018.01

This slides are licensed under a [Creative Commons Attribution-NonCommercial 4.0 International License](https://creativecommons.org/licenses/by-nc/4.0/).