Lab 12

Files and Persistence

Files and Directories in Java

Current working directory

- The directory from which you execute a Java Program

```
class Example {
    public static void main(String[] args) {
        System.out.println(System.getProperty("user.dir"));
    }
}
```

- Windows: C:\Users\Linda\example
- Mac: /home/Linda/example

Absolute Path vs Relative Path

- Absolute path: the canonical file path, relative to the root of the file system.
  - C:/Users/Linda/example/Example.java (Windows)
  - /home/Linda/example/Example.java (Mac/Linux)
- Relative path: the file path relative to the current working directory (CWD)
  - if we were in C:/Users/Linda/ (Windows) or /home/Linda/ (Mac/Linux), the relative file path would be example/Example.java.
- Common misunderstanding
  - The root of the file system (represented by /) is NOT the same as the home directory (represented by ~).

Java File Objects

- Represents a file **path**, NOT an actual file
  - Creating a java File object will NOT make the file actually exist
    - Calling .createnewFile() or .mkdir() will create a new file/dir
- Can use relative or absolute paths to create a File object

Persistence

Two ways of Achieving Persistence

1. Persisting Objects: Writing plain text to files
2. Persisting Objects: Serialization
    a. Why?
        i. Persisting complex, non-string objects
    b. Serialization: the process of converting objects to a series of bytes that can be stored in a file
        i. Can deserialize those bytes to get the object back

Demo.java

Utils.java

WARNING

- BE EXTREMELY CAREFUL WITH RM -RF
- COMMIT AND PUSH CHANGES OFTEN