Image Credit: Julius Schorzman

Java Lesson 4

Concept Review

Temporary Storage

- Temporary storage is commonly referred to as computer memory or RAM (Random Access Memory).

- This type of memory temporarily stores information while your program is running, but the moment that the application closes or the system loses power -- all information is immediately lost.

- This type of memory is also known as volatile.

Permanent Storage

- Permanent storage refers to the idea that an application can “write” data to a device, such as a hard drive, flash drive or even a CD that can potentially exist for considerably longer periods of time after a program has originally ended.

- Some devices that are considered permanent storage are:

  - Hard drives, Flash Drives, DVDs, CDs, Zip Drives, Diskettes, etc.

The File Class

- The **File** class comes with some additional features, which you may find useful with your programs, such as:
  - Determine the file's absolute path.
  - Rename a file / copy a file
  - Delete a file.
- **Quick reminder:** Any time that you're working with files, always be careful that you don't accidentally modify or delete anything important from your computer.

Absolute Path

- To determine a file's absolute path, use the getAbsolutePath() method. This will return a String that tells you exactly where the file is located.

```
import java.io.*;

public class DemoFiles
{
   public static void main(String[] args)
   {
      File myData = new File("quotes.txt");
      System.out.println("The absolute path is: " + myData.getAbsolutePath());
   }
}
```

Renaming and Moving Files

- Sometimes, you may encounter a situation where you need to rename a file or even move it to a new place on the drive. Luckily, in Java, it's very easy to do!

- Use the renameTo() method, which receives a new File object showing the desired location of where to move / rename the file.

```
import java.io.*;

public class DemoFiles
{
    public static void main(String[] args)
    {
        File myData = new File("quotes.txt");
        myData.renameTo(new File("C:\\Java\\Examples\\GreatQuotes.txt"));
    }
}
```

Deleting Files

- Java makes it very easy to delete files by using the delete() method. Check out the following example:

```
import java.io.*;

public class DemoFiles
{
   public static void main(String[] args)
   {
      File myData = new File("GreatQuotes.txt");
      myData.delete();
   }
}
```

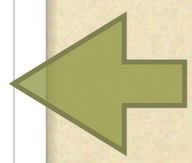
- **Caution:** Java does not warn the user that a file is about to be permanently erased. If you use the delete() method in regards to a file, it will simply proceed without warning the user. **Use with caution!**

File or Folder

- Sometimes, your application might need to know if a provided path is actually a file, or if it is a folder instead. The developers of Java realized this need, so they added a method called isDirectory() for you to use.

```
import java.io.*;

public class DemoFiles
{
    public static void main(String[] args)
    {
        File myData = new File("c:\\Windows\\");
        boolean aFolder = myData.isDirectory();
    }
}
```



List of Files

- What if you need to acquire a list of files in a directory? Java provides a way for developers to read the file system itself. Consider the following:

```
import java.io.*;

public class DemoFiles
{
   public static void main(String[] args)
   {
      File myData = new File("c:\\Java\\");
      String[] fileNames = myData.list();
   }
}
```

- Notice the list() method? It returns a String array with the names of current items that are stored in the specified File path.