1. Some languages are clearly compiled(e.g. C), and some clearly interpreted(e.g. Python). Where does Java fit into this model? How does Java use this model to provide some measure of machine independence? List the types of files you see in a project folder and how they relate to this.
2. What is the difference between “==” and equals() methods to compare two String objects?

“==” compares references (memory addresses) and checks identity equality. Equals() method compares values (content equality) Example:

String s1 = “text”; String s2 = s1; String s3 = “more text”; String s4 = “text”;

s2 == s1 //true checks identity equality, they both point to the same address

s1.equals(“text”); //true checks content equality

s1 == s4 //true checks identity equality, String Pool at work

1. Why is a String immutable>what does that mean? Give an example of an operation that demonstrates immutability and how you need to alter the code to work around it.

String is immutable in Java, that means an instance of the String cannot be modified. All information in an instance is initialized when the instance is created and the information cannot be modified. There are advantages of immutable classes. String is designed to be immutable.

* Requirement of String Pool. When a String is created, if the string already exists in the pool, the reference of the existing string will be returned instead of creating a new object. If the String were mutable, then changing one reference would lead to the wrong value for other references.
* Caching Hashcode – hash can be calculated once and cached so it does not have to be calculated everytime it is used, if the String could be modified, this hash would not be able to be cached
* Security - if String is used as a parameter for opening a file or a network connection etc, if it were changeable this could pose a security vulnerability.
* Immutable Objects are thread safe. Can be shared freely among multiple threads eliminating the need to do synchronization.
* Source https://www.programcreek.com/2013/04/why-string-is-immutable-in-java/

1. List the 8 fundamental primitive data types in Java. What are the key differences between the numeric data types on the list? String seems like a notable exception here; describe at least two ways we use strings that demonstrates their non-primitive nature.