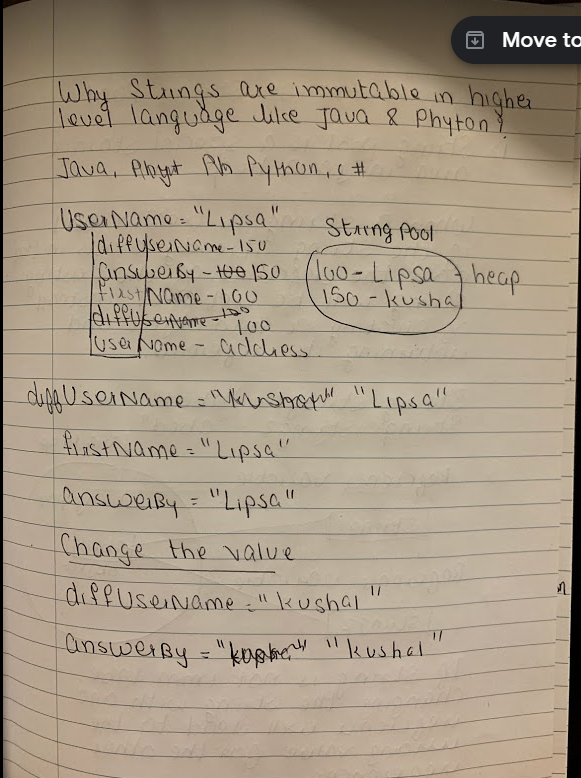
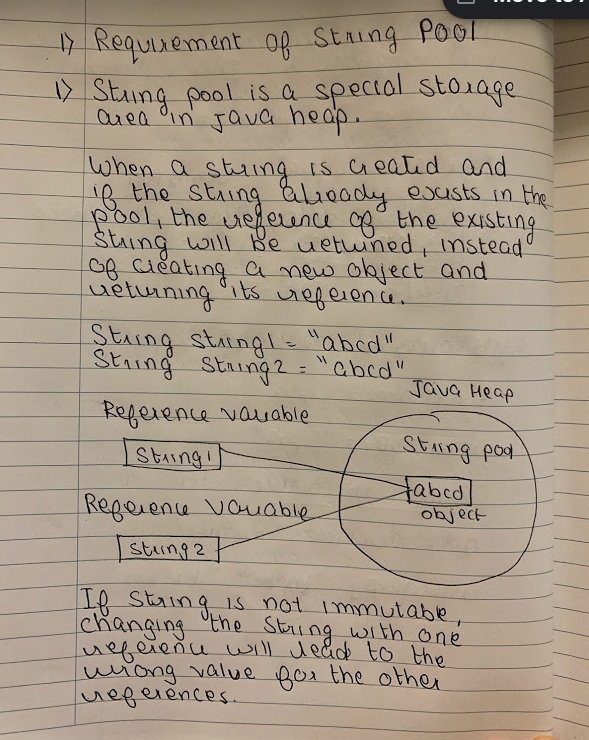
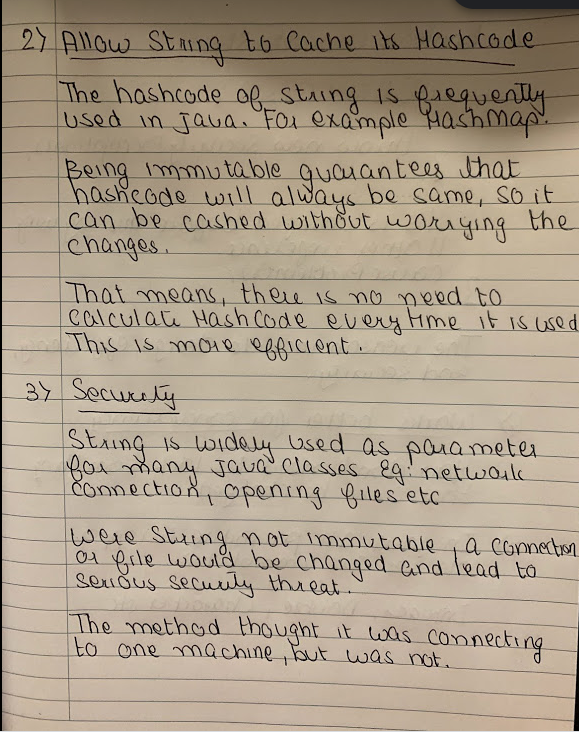
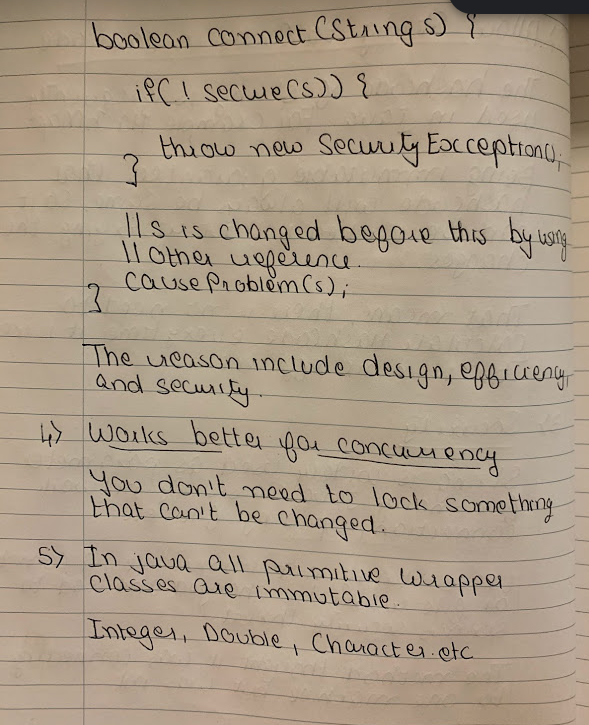
# Quiz 1:

1. Why Strings are immutable in higher programming language like Java, C# and Python?









# Question 1:



**Higher level languages like Java and Python keep strings immutable (why? Google for it). In these languages, what is the output of the code above?**

#### Select the correct choice:

"Interview Kickstart", because str was changed, and s points to it.

"Interview", because str now points to a new string literal and s still points to "Interview". String being immutable doesn't mean the reference to it is immutable.

This will give a runtime error, because we're trying to violate the most fundamental constraint of the language.

"Interview Kickstart", because immutability is an internal concept to the language and the programmers don't have to worry about it.

This will print empty string, because now that str is overwritten, due to immutability, s points to a something that doesn't exist.

# Answer 1:

"Interview", because str now points to a new string literal and s still points to "Interview". String being immutable doesn't mean the reference to it is immutable.

# Question 2:



**What will you use, in order to store a password - A String or Character Array?**

#### Select the correct choice:

String, because passwords are Strings.

String, because passwords need to be immutable.

Character Array, because Strings are immutable and can be dumped to see what the password is!

# Answer 2:

Character Array, because Strings are immutable and can be dumped to see what the password is!

1. **Strings are immutable:** Strings are immutable in Java and therefore if a password is stored as plain text it will be available in memory until Garbage collector clears it and as Strings are used in String pool for re-usability there are high chances that it will remain in memory for long duration, which is a security threat. Strings are immutable and there is no way that the content of Strings can be changed because any change will produce new String.  
   With an array, the data can be wiped explicitly after its work is complete. The array can be overwritten and and the password won’t be present anywhere in the system, even before garbage collection.
2. **Security:** Any one who has access to memory dump can find the password in clear text and that’s another reason to use encrypted password than plain text.  So Storing password in character array clearly mitigates security risk of stealing password.
3. **Log file safety:** With an array, one can explicitly wipe the data , overwrite the array and the password won’t be present anywhere in the system.  
   With plain String, there are much higher chances of accidentally printing the password to logs, monitors or some other insecure place. char[] is less vulnerable.

|  |
| --- |
| //Java program to illustate prefering char[] arrays  //over strings for passwords in Java  public class PasswordPreference  {      public static void main(String[] args)      {          String strPwd = "password";          char[] charPwd = new char[] {'p','a','s','s','w','o','r','d'};            System.out.println("String password: " + strPwd );          System.out.println("Character password: " + charPwd );      }  } |

Output:

String password: password

Character password: [C@15db9742