Summary
In this practical you write a program that determines if a password created by the user satisfies certain rules for a strong password. For this task you will use a sequence of if/else statements, including nested if/else statements. Finally, using the “git add”, “git commit”, and “git push” commands you will upload your source code to your Git repository hosted by Bitbucket.

Review the Textbook
You may refer to sections 5.1–5.2 in your textbook to read about if/else statements and boolean expressions.

Password Checker
You will write a program that will pre-validate the user’s password given certain requirements, if the user desires to do so. Your program should first ask if the user wants the password checked. If “yes”, then you should display a prompt to allow the user to enter the password. Then, using a sequence of if/else statements, you should proceed to check the password requirements stated below. If not, then you should display all the requirements, prompt the user for the password and save it. You should utilize a nested if/else structure to account for user’s choice for password pre-validation. A program with the initial statements to get you started is given in the course’s shared repository. Study the given program, then compile and run it before you proceed!

Password requirements:

1. A password can not contain any spaces. You can use contains() method of the String class to check if the user’s password contains a space (“ “).

2. A password must have at least eight characters. You can use length() method of the String class to check for the number of characters in the user’s password.

3. A password must contain at least one numeric digit. You can use contains() method of the String class and || boolean operator to check if the user’s password contains each of the ten possible digits.

4. A password must have at least one “-” or “_” character. You can use contains() method of the String class and || boolean operator to check if the user’s password contains at least one of these two allowable special characters.
5. A password must have at least one uppercase letter. You can use `equals` and `toLowerCase` methods of the `String` class and the boolean operator `!` to check if the password String given by the user is equal to the same string converted to all lowercase letters. If it is not equal, that means the original String was already in all lowercase letters and did not contain any uppercase letters. Assuming `password` is the String name for the user’s password, you can accomplish this by:
   ```java
   if(!password.equals(password.toLowerCase()))
   ```

6. A password must have at least one lowercase letter. You can use `equals` and `toUpperCase` methods of the `String` class and the boolean operator `!` to check if the password String given by the user is equal to the same string converted to all uppercase letters. If it is not equal, that means the original String was already in all uppercase letters and did not contain any lowercase letters. Assuming `password` is the String name for the user’s password, you can accomplish this by:
   ```java
   if(!password.equals(password.toUpperCase()))
   ```

Here is the sample output from running the program with the password `practical6` as input:

```
jjumadinova@aldenv113:~/practical6$ java PasswordChecker
Would you like to pre-validate your new password? yes
Please enter a password: practical6
Pass: Your password does not contain a space!
Pass: Your password has at least 8 characters!
Pass: Your password has at least one numeric digit!
Fail: Your password does not have at least one allowable special character!
Fail: Your password does not have at least one uppercase letter!
Pass: Your password has at least one lowercase letter!

jjumadinova@aldenv113:~/practical6$ java PasswordChecker
Would you like to pre-validate your new password? no
Password must meet these requirements:
   Must contain 8 characters.
   Must contain at least 1 numeric digit.
   Must contain at least 1 special character - or _.
   Must contain at least 1 upper case letter.
   Must contain at least 1 lower case letter.

Please enter a password: practical6
```

In addition to regularly compiling your program, you should thoroughly test it as well!

**Completing the Practical Assignment**

To finish this assignment and earn a “checkmark”, you should submit the `PasswordChecker.java` program in your Bitbucket repository by using appropriate `git` commands.