

EXPERIMENT NO. 15

1. What is Regular Expression? What is use of Regular Expression?

Ans:

Java provides the `java.util.regex` package for pattern matching with regular expressions. Java regular expressions are very similar to the Perl programming language and very easy to learn. A regular expression is a special sequence of characters that helps you match or find other strings or sets of strings, using a specialized syntax held in a pattern. They can be used to search, edit, or manipulate text and to define constraint on strings such as password

2. How to validate the data using Regular Expression in java?

Ans:

The `java.util.regex` package consists following three classes-
Pattern Class – A Pattern object is a compiled representation of a regular expression. To create a pattern, you must first invoke one of its public static `compile()` methods, which will then return a Pattern object. These methods accept a regular expression as the first argument.

Matcher Class – A Matcher object is the engine that interprets the pattern and performs match operations against an input string. You obtain a Matcher object by invoking the `matcher()` method on a Pattern object.

Example-Here this regular expression allows String that must contain one digit, one lower case char, one upper case char, some special chars, length should be within 6 to 15 chars.

```
private static Pattern pswNamePtrn =
Pattern.compile("((?=.*\\d)(?=.*[a-z])(?=.*[A-Z])(?=.*[@$%]).{6,15})");

public static boolean validatePassword(String userName){

    Matcher mtch = pswNamePtrn.matcher(userName);
    if(mtch.matches()){
        return true;
    }
    return false;

    System.out.println("Is 'gabbarsingh' a valid password?" + validatePassword("gabbarsingh"));

    output: Is 'gabbarsingh' a valid password? false
```



3. What are the Classes and interfaces are given by java.util.regex package?

Ans:

- 1.MatchResult interface
- 2.Matcher class
- 3.Pattern class
- 4.PatternSyntaxException class

Pattern Class — A Pattern object is a compiled representation of a regular expression. To create a pattern, you must first invoke one of its public static `compile()` methods, which will then return a Pattern object. These methods accept a regular expression as the first argument.

Matcher Class — A Matcher object is the engine that interprets the pattern and performs match operations against an input string. You obtain a Matcher object by invoking the `matcher()` method on a Pattern object.

PatternSyntaxException — A PatternSyntaxException object is an unchecked exception that indicates a syntax error in a regular expression pattern.

MatchResult interface — This interface contains query methods used to determine the results of a match against a regular expression. The match boundaries, groups and group boundaries can be seen but not modified through a MatchResult



4. Write a code to validate Student PRN No., Name, Mobile no and Age using regular expression in java.

Ans:

```
#Name
public static boolean validateLetters(String txt) {
String regx = "[\\p{L} .'-]+";
Pattern pattern = Pattern.compile(regx,Pattern.CASE_INSENSITIVE);
Matcher matcher = pattern.matcher(txt);
return matcher.find();
}

#age
public static boolean validateLetters(String txt) {
String regx = "(0?[1-9]|[1-9][0-9]|[1][1-9][1-9]|200)$";
Pattern pattern = Pattern.compile(regx,Pattern.CASE_INSENSITIVE);
Matcher matcher = pattern.matcher(txt);
return matcher.find();
}

#Phone
public static boolean validateLetters(String txt) {
String regx = "[789]\\d{9}$";
Pattern pattern = Pattern.compile(regx,Pattern.CASE_INSENSITIVE);
Matcher matcher = pattern.matcher(txt);
return matcher.find();
}

#PRN|
public static boolean validateLetters(String txt) {
String regx "\\d{2}[a-z]{4,}\\d{2}"; //15inft29 or 15dsinft29 formats
Pattern pattern = Pattern.compile(regx,Pattern.CASE_INSENSITIVE);
Matcher matcher = pattern.matcher(txt);
return matcher.find();
}
```

