

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ



# Introduction

**Dr. Nasir Jalal PhD (Computer Science)**

**Lecturer**

**CS & IT Department, CUVAS  
Bahawalpur**

# Course Detail



## BS CS 5<sup>th</sup> Semester Fall 2022-2026

Course: Theory of Programming Languages

Course Code: CS IT 507

## Lecture: 6 (20-09-2024)

# Outline Lecture 5



- Revision of Lecture 5
- Regular Expression ....continue...

# Regular Expression



A regular expression (also called regex or regexp) is a way to describe a pattern. It is used to locate or validate specific strings or patterns of text in a sentence, document, or any other character input.

# characters used in Regular Expression



Character	Full Name	Regex Type	Description
?	Question Mark	Quantifier	Matches zero or one preceding character.
*	Asterisk	Quantifier	Matches zero or more preceding characters.
+	Plus Sign	Quantifier	Matches one or more preceding characters.
\	Backslash	Escape Character	BRE: Indicates the proceeding character is special. ERE: Indicates the proceeding character is basic.
[ ]	Square Brackets	Grouping	Creates a character group or range.
( )	Parentheses	Grouping	Creates a sequence or sub-expression.

# characters used in Regular Expression



Character	Full Name	Regex Type	Description
{ }	Curly Braces	Quantifier	Creates a specific numerical quantifier range.
^	Caret	Assertion (Anchor)	Matches the beginning of a line.
\$	Dollar Sign	Assertion (Anchor)	Matches the end of a line.
\b	Word Boundary	Assertion (Anchor)	Matches a word boundary (a non-word character such as a space, tab, or period).
.	Period	Shorthand	Matches any single character.
	Pipe	Alternation	Logical OR operator.



# Regular Expression

Two types of Regular Expression

- BRE (Basic Regular Expression)
- ERE (Extended Regular Expression)

The difference between ERE and BRE is what happens when you add a backslash in front of a character within an expression.

Backslash Status	BRE	ERE
Present	The character is special.	The character is basic.
Not Present	The character is basic.	The character is special.



# Why Assertion necessary



Assertions are special characters in regular expressions that remove ambiguity or partial matching from an expression.

Character	Full Name	Regex Type	Description
^	Caret	Assertion (Anchor)	Matches the beginning of a line.
\$	Dollar Sign	Assertion (Anchor)	Matches the end of a line.

# Why Assertion necessary



RE without Assertion(anchor)

***[a-z]***

Since there are no assertion characters in the expression, the input *123word!* would match the expression because it includes letters within the *a-z* range: w, o, r, or d.

RE with Assertion(anchor)

***^[a-z]\$***

*Now the input 123word does not match*

# Assertion Example



- We want to use regex to validate a JavaScript form on a website to ensure only real email addresses are entered. An email address must have an @ symbol and at least a two-letter TLD (top-level domain) like .co or .com, as shown below.

**`^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$`**

# Assertion Example Cont... ..



**`^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$`**

- Any combination of lowercase and uppercase letters and numbers, as well as periods, underscores, percent signs, plus signs, and minus signs.
- The plus sign outside the brackets indicates that one or more preceding grouped characters should be matched.
- An @ sign. The expression wants to match this character exactly, so it is not in a group.
- Any combination of lowercase and uppercase letters and numbers, as well as periods and minus signs.
- A period. As a period is a special character in ERE, we must escape it (include a backslash), so we can match the period character exactly.
- A minimum of two uppercase and/or lowercase letters. The curly braces quantify the group to 2 or more matches.

# Assertion Question



To verify valid credit card numbers have been input into a system. Credit card numbers begin with either 25 or 26 and are 13 digits long.

**`^2[56][0-9]{11}$`**

# THANKS