

# Computing Fundamentals

## Planning

Session	Subject	Test – Hand-in
1	Network Models	
2	Internet Protocol Suite	
3	Network segmentation	
4	Network protocols	
5	Operating systems	
6	Command Line	
-- 30/10 – 5/11 --	Autumn break – HERFSTVAKANTIE	
7	Command Line	
8	Mid-term test	<b>Test</b>
9	<b>Scripting</b>	
10	Virtualization - Cloud computing	

# Computing Fundamentals

---

## Command-line interface (CLI)

- A command-line interface (CLI) is a means of interacting with a computer program by inputting lines of text called command-lines. Command-line interfaces emerged in the mid-1960s, on computer terminals, as an interactive and more user-friendly alternative to the non-interactive interface available with punched cards.
- Today, most computer users rely on graphical user interfaces ("GUIs") instead of CLIs. However, many programs and operating system utilities lack GUIs, and are intended to be used through CLIs.

# Computing Fundamentals

---

## Command-line interface (CLI)

- Knowledge of CLIs is also useful for writing scripts. Programs that have CLIs are generally easy to automate via scripting, since command-lines, being mere lines of text, are easy to specify in code.
- CLIs are made possible by command-line interpreters or command-line processors, which are programs that read command-lines and carry out the commands.

# Computing Fundamentals

---

## Types

- **Operating system command-line interfaces**

Operating system (OS) command-line interfaces are usually distinct programs supplied with the operating system. A program that implements such a text interface is often called a command-line interpreter, command processor or shell.

- **Application command-line interfaces**

Application programs (as opposed to operating systems) may also have command-line interfaces.

# Computing Fundamentals

---

## Application command-line interfaces

- An application program may support none, any, or all of these three major types of command-line interface mechanisms:
- **Parameters:** Most command-line interfaces support a means to pass additional information to a program when it is launched.
- **Interactive command-line sessions:** After launch, a program may provide an operator with an independent means to enter commands.
- **Inter-process communication:** Most operating systems support means of inter-process communication (for example, standard streams or named pipes). Command lines from client processes may be redirected to a CLI program by one of these methods.

# Computing Fundamentals

---

## NOS command-line interface (CLI)

- NOS CLIs like **Cisco**, **Meraki**, **Ruckus**, ..., are the primary user interface used for configuring, monitoring, and maintaining network devices. The user interface allows you to directly and simply execute commands, whether using a router console or terminal, or using remote access methods.
- Additional user interfaces include **Setup mode** (used for first-time startup), the Cisco Web Browser, and user menus configured by a system administrator.

# Computing Fundamentals

---

## NOS command-line interface (CLI)

- NOS CLIs like **Cisco**, **Meraki**, **Ruckus**, ..., are the primary user interface used for configuring, monitoring, and maintaining network devices. The user interface allows you to directly and simply execute commands, whether using a router console or terminal, or using remote access methods.
- Additional user interfaces include **Setup mode** (used for first-time startup), the Web Browser, and user menus configured by a system administrator.



# Computing Fundamentals

---

## Cisco IOS XE CLI Task List

- To familiarize yourself with the features of the Cisco IOS XE CLI, perform any of the tasks described in the following sections:
- Getting Context-Sensitive Help
- Using the no and default Forms of Commands
- Using Command History
- Using CLI Editing Features and Shortcuts
- Searching and Filtering CLI Output

# Computing Fundamentals

---

## Cisco IOS XE CLI Task List

- To familiarize yourself with the features of the Cisco IOS XE CLI, perform any of the tasks described in the following sections:
- Getting Context-Sensitive Help
- Using the no and default Forms of Commands
- Using Command History
- Using CLI Editing Features and Shortcuts
- Searching and Filtering CLI Output

# Computing Fundamentals

---

## Cisco IOS XE CLI

- Getting Context-Sensitive Help

Command	Purpose
(prompt )# <b>help</b>	Displays a brief description of the help system.
(prompt )# abbreviated-command-entry?	Lists commands in the current mode that begin with a particular character string.
(prompt )# abbreviated-command-entry < <b>Tab</b> >	Completes a partial command name.
(prompt )# <b>?</b>	Lists all commands available in the command mode.
(prompt )# command?	Lists the available syntax options (arguments and keywords) for the command.
(prompt )# command keyword <b>?</b>	Lists the next available syntax option for the command.

# Computing Fundamentals

## Cisco IOS XE CLI

- Key Combinations Used to Move the Cursor

Keystrokes	Function Summary	Function Details
Left Arrow or Ctrl -B	B ack character	Moves the cursor one character to the left. When you enter a command that extends beyond a single line, you can press the Left Arrow or Ctrl-B keys repeatedly to scroll back toward the system prompt and verify the beginning of the command entry, or you can press the Ctrl-A key combination.
Right Arrow or Ctrl -F	F orward character	Moves the cursor one character to the right.
Esc , B	B ack word	Moves the cursor back one word.
Esc , F	F orward word	Moves the cursor forward one word.
Ctrl -A	Beginning of line	Moves the cursor to the beginning of the line.
Ctrl -E	E nd of line	Moves the cursor to the end of the command line.

# Computing Fundamentals

---

## Cisco IOS XE CLI

- **Completing a Partial Command Name**

If you cannot remember a complete command name, or if you want to reduce the amount of typing you have to perform, enter the first few letters of the command, then press the Tab key. The command line parser will complete the command if the string entered is unique to the command mode. If your keyboard does not have a Tab key, press Ctrl -I instead.

- The CLI will recognize a command once you have entered enough characters to make the command unique. For example, if you enter conf in privileged EXEC mode, the CLI will be able to associate your entry with the configure command, because only the configure command begins with conf .
- In the following example the CLI recognizes the unique string for privileged EXEC mode of conf when the Tab key is pressed:
- **Router# conf**
- **<Tab>**
- **Router# configure**

# Computing Fundamentals

---

## Cisco IOS XE CLI

- Editing Command Lines that Wrap

The CLI provides a wrap-around feature for commands that extend beyond a single line on the screen. When the cursor reaches the right margin, the command line shifts ten spaces to the left. You cannot see the first ten characters of the line, but you can scroll back and check the syntax at the beginning of the command. To scroll back, press Ctrl-B or the Left Arrow key repeatedly until you scroll back to the beginning of the command entry, or press Ctrl-A to return directly to the beginning of the line. In the following example, the access-list command entry extends beyond one line. When the cursor first reaches the end of the line, the line is shifted ten spaces to the left and redisplayed. The dollar sign (\$) indicates that the line has been scrolled to the left. Each time the cursor reaches the end of the line, the line is again shifted ten spaces to the left.

- Router(config)# access-list 101 permit tcp 172.41.134.5 255.255.255.0 172.41.1
- Router(config)# \$ 101 permit tcp 172.41.134.5 255.255.255.0 172.41.135.0 255.255
- Router(config)# \$t tcp 172.41.134.5 255.255.255.0 172.31.135.0 255.255.255.0 eq
- Router(config)# \$41.134.5 255.255.255.0 172.41.135.0 255.255.255.0 eq 35

# Computing Fundamentals

---

## Cisco IOS XE CLI

- **Deleting Entries**

Use any of the following keys or key combinations to delete command entries if you make a mistake or change your mind:

Keystrokes	Purpose
<b>Delete</b> or <b>Backspace</b>	Deletes the character to the left of the cursor.
<b>Ctrl -D</b>	Deletes the character at the cursor.
<b>Ctrl -K</b>	Deletes all characters from the cursor to the end of the command line.
<b>Ctrl -U</b> or <b>Ctrl -X</b>	Deletes all characters from the cursor to the beginning of the command line.
<b>Ctrl -W</b>	Deletes the word to the left of the cursor.
<b>Esc , D</b>	Deletes from the cursor to the end of the word.

# Computing Fundamentals

## Cisco IOS XE CLI

- Configuring Global Parameters

To configure global parameters for your router, follow these steps:

### Procedure

	Command or Action	Purpose
Step 1	<b>configure terminal</b> <b>Example:</b>  Router> <b>enable</b> Router# <b>configure terminal</b> Router(config)#	Enters global configuration mode when using the console port.  Use the following to connect to the router with a remote terminal:  telnet router-name or address Login: login-id Password: ***** Router> enable
Step 2	<b>hostname</b> <i>name</i> <b>Example:</b>  Router(config)# <b>hostname Router</b>	Specifies the name for the router.
Step 3	<b>enable password</b> <i>password</i> <b>Example:</b>  Router(config)# <b>enable password crlny5ho</b>	Specifies a password to prevent unauthorized access to the router.  <b>Note</b> In this form of the command, password is not encrypted. To encrypt the password use enable secret password as noted in the previously mentioned Device Hardening Guide.



# Computing Fundamentals

## Cisco IOS XE CLI

- Configuring a Loopback Interface

The loopback interface acts as a placeholder for the static IP address and provides default routing information. To configure a loopback interface, follow these steps:

### Procedure

	Command or Action	Purpose
Step 1	<b>interface</b> <i>type number</i> <b>Example:</b>  Router(config)# <b>interface Loopback 0</b>	Enters configuration mode on the loopback interface.
Step 2	(Option 1) <b>ip address</b> <i>ip-address mask</i> <b>Example:</b>  Router(config-if)# <b>ip address 10.108.1.1 255.255.255.0</b>	Sets the IP address and subnet mask on the loopback interface. (If you are configuring an IPv6 address, use the <b>ipv6 address</b> <i>ipv6-address/prefix</i> command described below.
Step 3	(Option 2) <b>ipv6 address</b> <i>ipv6-address/prefix</i> <b>Example:</b>  Router(config-if)# <b>ipv6 address 2001:db8::ffff:1/128</b>	Sets the IPv6 address and prefix on the loopback interface.
Step 4	<b>exit</b> <b>Example:</b>  Router(config-if)# <b>exit</b>	Exits configuration mode for the loopback interface and returns to global configuration mode.

# Computing Fundamentals

---

## Cisco IOS XE CLI

- Editing Command Lines that Wrap

The CLI provides a wrap-around feature for commands that extend beyond a single line on the screen. When the cursor reaches the right margin, the command line shifts ten spaces to the left. You cannot see the first ten characters of the line, but you can scroll back and check the syntax at the beginning of the command. To scroll back, press Ctrl-B or the Left Arrow key repeatedly until you scroll back to the beginning of the command entry, or press Ctrl-A to return directly to the beginning of the line. In the following example, the access-list command entry extends beyond one line. When the cursor first reaches the end of the line, the line is shifted ten spaces to the left and redisplayed. The dollar sign (\$) indicates that the line has been scrolled to the left. Each time the cursor reaches the end of the line, the line is again shifted ten spaces to the left.

- Router(config)# access-list 101 permit tcp 172.41.134.5 255.255.255.0 172.41.1
- Router(config)# \$ 101 permit tcp 172.41.134.5 255.255.255.0 172.41.135.0 255.255
- Router(config)# \$t tcp 172.41.134.5 255.255.255.0 172.31.135.0 255.255.255.0 eq
- Router(config)# \$41.134.5 255.255.255.0 172.41.135.0 255.255.255.0 eq 35

# Computing Fundamentals

---

## Cisco IOS XE CLI

- Editing Command Lines that Wrap

The CLI provides a wrap-around feature for commands that extend beyond a single line on the screen. When the cursor reaches the right margin, the command line shifts ten spaces to the left. You cannot see the first ten characters of the line, but you can scroll back and check the syntax at the beginning of the command. To scroll back, press Ctrl-B or the Left Arrow key repeatedly until you scroll back to the beginning of the command entry, or press Ctrl-A to return directly to the beginning of the line. In the following example, the access-list command entry extends beyond one line. When the cursor first reaches the end of the line, the line is shifted ten spaces to the left and redisplayed. The dollar sign (\$) indicates that the line has been scrolled to the left. Each time the cursor reaches the end of the line, the line is again shifted ten spaces to the left.

- Router(config)# access-list 101 permit tcp 172.41.134.5 255.255.255.0 172.41.1
- Router(config)# \$ 101 permit tcp 172.41.134.5 255.255.255.0 172.41.135.0 255.255
- Router(config)# \$t tcp 172.41.134.5 255.255.255.0 172.31.135.0 255.255.255.0 eq
- Router(config)# \$41.134.5 255.255.255.0 172.41.135.0 255.255.255.0 eq 35