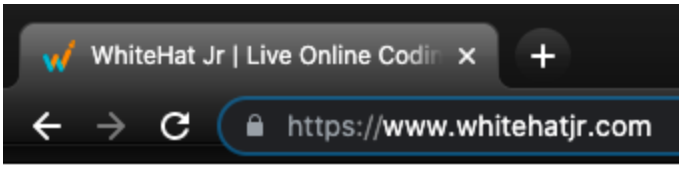
Topic	PYTHON & PORTS	
Class Description	**Students will revise some of the commonly used concepts of Python and understand a bit more about ports.**	
Class	**C-198**	
Class time	**45 mins**	
Goal	• Revise the common concepts of Python • Understand about ports in software development	

Resources Required	• Teacher Resources:
	  ○ Google Colab
	  ○ laptop with internet connectivity
	  ○ earphones with mic
	  ○ notebook and pen
	• Student Resources:
	  ○ Google Colab
	  ○ laptop with internet connectivity
	  ○ earphones with mic
	  ○ notebook and pen	

Class structure	**Warm Up** **Teacher - led Activity 1** **Student - led Activity 1** **Wrap up**	**5 mins** **25 mins** **10 mins** **5 min**
WARM UP SESSION - 5 mins		
CONTEXT • **Learning about different kinds of ports** • **Revisiting Python concepts**		

Teacher Action	Student Action
Hey How have you been ? It's great to see you again! Remember what we learned in the previous Class? Are you excited to learn something new today? Today we are going to learn about Ports and along with that will revise the python concept too.	**ESR:** I am good. We learnt about how to set up a Server and Client **and** learnt about Troubleshooting
Q&A Session	
Question	**Answer**
High speed ethernet works on which type of cable? A.Coaxial Cable B.Copper twisted pair Cable C.Optical Fibre D.Straight Cable	**C**
File Transfer protocol (FTP) is built on which architecture? A.Peer to Peer B.FTP C.Client-Server D.All of the above	**C**
TEACHER-LED ACTIVITY - 25 mins	

Teacher Initiates Screen Share	
CHALLENGE • **Understanding Ports** • **Revisiting Python Concepts**	
Teacher Action	**Student Action**
We all have been surfing so much on the internet everyday and we have even come across many websites. Most of the websites open with either an **“http://”** while some open with an **“https://”**. Have you ever noticed this difference ?	**ESR:** HTTP is the “Hypertext Transfer Protocol” and HTTPS is the “Hypertext Transfer Protocol Secure”.
Great! You know this. Let’s understand more about this Now if you would open any website, you would notice that the browser automatically specifies an “HTTP” or an “HTTPS” before it. Try typing just **“whitehatjr.com/”** in a new tab and observe the URL once the website opens *Teacher tries it in a new tab*	*Student tries it in a new tab*
 And when you press enter, the URL automatically translates to -	

<table>
<tr><td colspan="2"> 
</td></tr>
<tr><td>You see how the browser automatically inserted an “https://” before the URL?

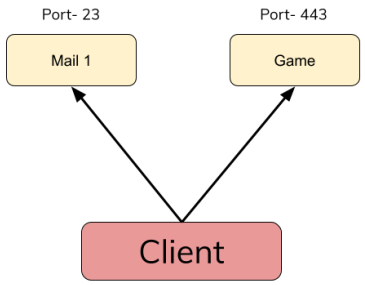
How does it happen?</td><td>ESR:
Varied!</td></tr>
<tr><td>It happens because WhitehatJr’s web server is an HTTPS server which means secure server , and it's fetching all the incoming connections on a particular port address i.e whitehatjr.com in our case.

It means as soon as we enter “whitehatjr.com” in our browser, the HTTPS server running in the backend will detect our incoming request and in response send the HTML content of the website through a port on your browser.HTTP generally use Transmission Control Protocol (TCP) connections to communicate through port

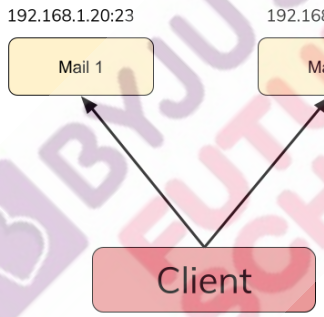
Now you must be wondering what is this TCP andPort

Let’s have better understanding on TCP/Port</td><td>ESR:
Varied!</td></tr>
</table>

TCP is a Transmission Control Protocol which is used to build a connection between the source computer and the destination one on the network.For devices to communicate via TCP, they use TCP ports. Every port assigned with a unique number	
Every port has unique number based on different different applications: Let's understand this with an example. Suppose you've opened 2 different applications on your device One is Email application and second one is video game. Now the Email application and Video game application can be transferring data to the server at the same time and it's happening simultaneously. Can you tell me how it might be doing so? How do computers know which application to send at which TCP/Port? As we discussed earlier,In order to differentiate between ports based on applications different unique numbers are assigned to these applications. Therefore we have number of ports available in the network	**ESR:** It is sending data to a different TCP/Port for email application and different TCP/Port for video games.



Now I am damm sure you must be wondering what if you're running the same type of 2 Same type of applications? What do you think the port number would be the same? Right? Let's see how it will work ?	**ESR:** Varied!
That's where IP Addresses come into play. A port number is always used along with an IP address. Both IP address and port number are used in the following way - ***ip_address:port_number*** Consider the following example -	

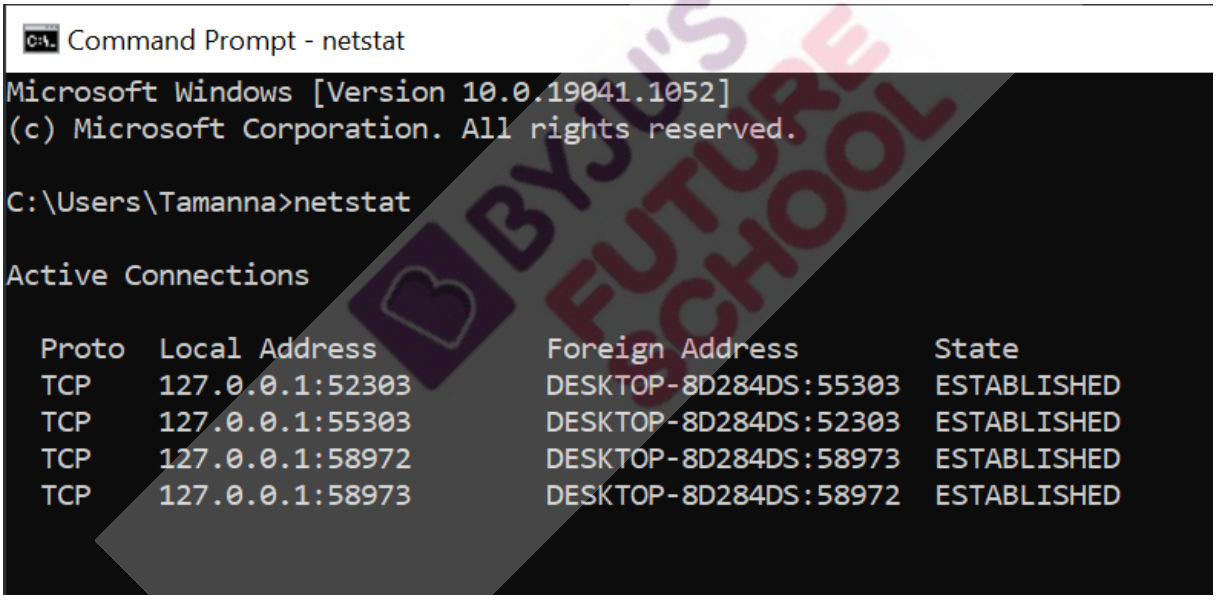
192.168.0.166:2691 Can you guess what the IP Address and the port number is in this case? Excellent! It means if two same applications will be on the same port but their IP address varies, which makes their combination unique, they can run simultaneously.	**ESR:** 192.168.0.166 is the IP address and 2691 is the port number.
	
A TCP port is a 16 bit unsigned value. This means that there can only be a finite number of TCP ports available in the world. Can you explain a bit? Great!	**ESR:** Bit is binary Digit

Bit is a Binary digits that can hold only one of two values: 0 or 1. That's why we are taking 2 as base and 16 as exponential power which results in 65536 values. But we consider from 0, that's why we have specifically 65,535 ports available in the world. Now out of these 65,535 ports, only the first 1,024 ports are well known ports majorly agreed upon in the world of technology for specific applications For example, the well known port of HTTP is 80. Similarly, port 443 is well known for HTTPS and it used to transfer data securely. Let's have a look at what different ports means:	
Teacher refers to Teacher Activity 1 and opens it in a new tab.	*Student refers to Student Activity 1 and opens it in a new tab.*
Just scroll down a bit and take a look here, you will see different port numbers and what they are used for -	

Most common used ports:
Teacher will try to explain the above mentioned ports along with their applications

Port No	Process Name	Description	Application
20/21	FTP	File Transfer Protocol	Data Transfer
23	SMTP	Simple Mail Transport Protocol	Email Routing
80	HTTP	HyperText Transfer Protocol	Used to transfer hypertext such as web pages and share data
443	HTTPs	HyperText Transfer Protocol Secure	Extension of HTTP Protocol, used for secure communication over Protocol
53	DNS	Domain Name Server	Used to establish connection between web servers and web sites
22	SSH	Secure Shell	Enables two computers to communicate
123	NTP	Network Time Protocol	Used for synchronizing multiple networks
23	Telnet	Telnet	Used for Server-Client Program
110	POP3	Post Office Protocol	Help you to download message/email from you inbox to Local Computer

Now, You must be curious about which of your applications uses which port number? Right ? Let's do one Activity *Teacher Activity 2:* • *Open a CMD prompt* • *Type in the command:* ***netstat*** *Command will be the same for both Windows and Mac operating systems. Guide the student to do the same.*	*Student Activity 2*



Ports other than 1024 i.e in the range 1024 to 49151 are reserved for future ports. And if you look at ports above in the range i.e 49152 to 65535 they are unofficially used by different technologies.	

Example:Remember how our Flask server always runs on port 5000 by default?

```
* Serving Flask app "app" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: off
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

Look closely at the last line. It says that it's running on ***127.0.0.1:5000.***

Here, ***127.0.0.1*** is the IP Address and ***5000*** is the port number. They are separated by a ***:***

If we run another Flask server while our first one is running, we get -

```
    self.socket.bind(self.server_address)
OSError: [Errno 48] Address already in use
```

Here, you will notice that it gives an error which says that ***Address is already in use***

This is because our second Flask App is again trying to run on IP address ***127.0.0.1*** and on port number ***5000***. This combination of IP Address and Port is already in use, therefore it says that the address is already in use.

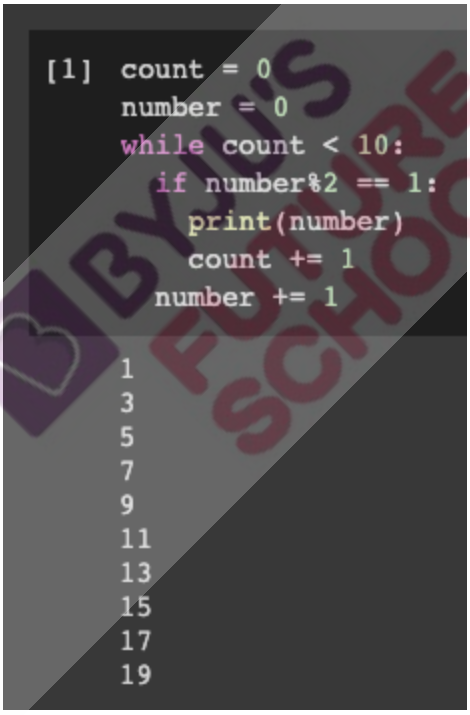
When we run a Flask app, we notice that it only runs when we add a port **:5000** to it in the URL.

That's because as soon as we enter it, our browser knows that it has to contact port 5000 on a domain known as ***localhost*** or ***127.0.0.1*** since our server

runs locally and not on a remote server which has a domain. Now you are wondering if all computers/ websites/servers have a local ip address but we remember domain names like google.com, whitehatjr.com, amazon. com.Actually in backend domain names are usually mapped with IP Addresses of the server because it's easy -to remember domain names since they are easier to remember as compared to complex IP addresses. So people usually access websites with domain names.	
Now we learnt about ports as well as IP addresses. When we combine IP addresses and ports together it becomes a **socket**. When we enter any website, let's say **"https://www.whitehatjr.com",** we are simply just connecting with the IP Address or domain name that whitehatjr.com and it is mapped to port i.e https means that we are connecting with it through port 443. Will learn more about socket and socket programming in the next class, and deep dive into what sockets are, but now will revise some concepts of Python	
When we learnt about Python, we went over -	**ESR:** Python has • Integers

• Input • Variables • Data Types Can you tell me what data types python has?	• Strings • Booleans • None • Float
Great! Do you also remember what lists are in Python?	**ESR:** Lists are array-like data structures in Python that can hold any number of values regardless of their data types. The elements are separated with a comma (,) and are placed within square brackets [].
Awesome. Now all elements in array have index value The first element of a list has index 0. The second element has an index 1. Third element has an index 2. and so on. Now what would be the index of the 8th element in the list?	**ESR:** 7
We had also learnt about methods in list 1. **len(list)** to find the length of the lists. 2. **list.append(element)** to add an element to the lists.	**ESR:**
We had also learnt about dictionaries in Python! Can you recall that ?	**ESR:** Dictionaries are similar to Objects in Python. They have key and value pairs that can hold information for us.

Teacher opens a new Google Colab from Teacher Activity 2	*Student opens a new Google Colab from Student Activity 2*
Consider a problem statement where we have to print the first 10 odd numbers using a while loop in Python. Can you try and help me write the code for it in a new Google Colab? *Teacher writes the code as student guides her*	*Student guides the teacher*



```
[1] count = 0
    number = 0
    while count < 10:
      if number%2 == 1:
        print(number)
        count += 1
      number += 1

1
3
5
7
9
11
13
15
17
19
```

Here we need to create two variables, one is for Count to maintain the count of numbers and one variable Number to print numbers whether it is even or odd.

Then we need to create a **while** loop where we place the condition of ***count < 10***. This means that this while loop will keep executing until our count is less than 10.

Inside it, with the help of an ***if condition*** we check if the number would be divisible by 2

<table>
<tr><td colspan="2">or not with the % operator. If it’s not (in case, it’s an odd number), we will print the number and increase the count by 1 to proceed further.

Outside the if condition, we are increasing the number by 1.</td></tr>
<tr><td>Can you tell me why we are increasing our number by 1 outside the if condition?</td><td>ESR:
It’s because we want to increase the number regardless of if it’s divisible by 2 or not. If we place it inside the if condition, it will only increase by 1 when the condition is true.</td></tr>
<tr><td>Great!

Now let’s consider another problem where we have to create a function which takes a dictionary and prints 2 lists. First list would contain all the keys and the second would contain all the values.

Teacher writes the code as student guides her</td><td>Student guides the teacher</td></tr>
<tr><td colspan="2">

```
[2] def print_dict(dictionary):
      keys = []
      values = []
      for key, value in dictionary.items():
        keys.append(key)
        values.append(value)
      print(keys)
      print(values)
```

The output -</td></tr>
</table>

```
[4] dict_1 = {
        "apples": 1,
        "bananas": 5,
        "oranges": 10
    }

    print_dict(dict_1)

    ['apples', 'bananas', 'oranges']
    [1, 5, 10]
```

Here, we created a function ***print_dict()*** that takes an argument ***dictionary***.

Inside this function, we create 2 variables - ***keys and values***. Both of these are empty lists to store all the keys and values of the dictionary that we receive in the argument.

Next, we have a ***for loop*** in which, with the help of ***items()*** method of a dictionary, we are getting both ***key, value*** in the variables.

Inside the loop, we are using the ***append()*** method to add the key and value to their corresponding lists we created earlier.

Finally, we are printing the result.

Alright, now it's time for you to practice some coding in Python!	

Teacher Stops Screen Share

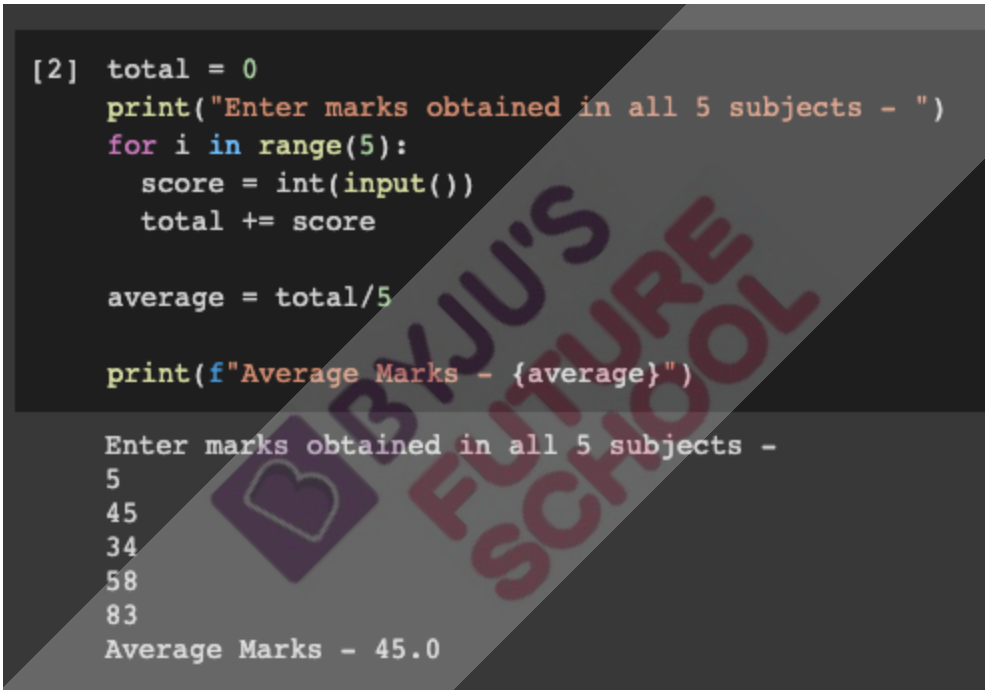
STUDENT-LED ACTIVITY - 10 mins

- **Ask the student to press the ESC key to come back to the panel.**
- **Guide the student to start Screen Share.**
- **The teacher gets into Fullscreen.**

ACTIVITY

- **Consider a problem statement where you got results from your school and now you need to find the average of marks scored in all subjects. There will**

be 5 subjects and the marks will be taken as an input from the user. The marks for each subject are out of 100.	
Teacher Action	**Student Action**
Teacher guides the student to write a average program	*Student writes the program to calculate the average score of 5 subjects*



```
[2] total = 0
    print("Enter marks obtained in all 5 subjects - ")
    for i in range(5):
      score = int(input())
      total += score

    average = total/5

    print(f"Average Marks - {average}")

    Enter marks obtained in all 5 subjects -
    5
    45
    34
    58
    83
    Average Marks - 45.0
```

The way average is calculated is that you take out the sum of all the values, and then divide it by the total number of values.

In our case, we know that the total number of values is 5. How we can start it is that we create a variable ***total*** to store the sum of all the inputs from the user.

Next, we create a ***for*** loop and iterate it 5 times, inside which we first take an input from the user, convert it into an integer with the ***int()*** function and then finally add it to the ***total.***

Lastly, we create a variable ***average*** in which we divide our ***total*** by ***5***, just as what the

<table>
<tr><td colspan="2">formula says.

Finally, we are printing the average score.</td></tr>
<tr><td colspan="2">ADDITIONAL STUDENT ACTIVITY - 4
● Consider a problem statement where you have to print a python program to make a simple calculator</td></tr>
<tr><td>Teacher guides the student to write a program for calculator</td><td>Student writes the code for the calculator program</td></tr>
</table>

```
print("CALCULATOR")
print("1. Addition")
print("2. Subtraction")
print("3. Multiplication")
print("4. Division")
choice = int(input("Enter your choice (1, 2, 3, 4): "))
if (choice >= 1 and choice <= 4):
  print("Enter two numbers: ")
  number_1 = int(input("Enter the first number - "))
  number_2 = int(input("Enter the second number - "))

  if choice == 1:
    print(f"Result - {number_1 + number_2}")

  elif choice == 2:
    print(f"Result - {number_1 - number_2}")

  elif choice == 3:
    print(f"Result - {number_1 * number_2}")

  else:
    print(f"Result - {number_1 / number_2}")

else:
  print("Incorrect Input!")
```

```
CALCULATOR
1. Addition
2. Subtraction
3. Multiplication
4. Division
Enter your choice (1, 2, 3, 4): 1
Enter two numbers:
Enter the first number - 33
Enter the second number - 88
Result - 121
```

Since we are building a calculator app, the first thing that we want to do is to provide the user with a menu to choose from.

Our first couple of lines of code is to print the menu.

Next, we ask the user to choose an option from 1 to 4. We take it as an ***input()*** and convert it into an integer with the ***int()*** function. Now we check, with the help of an ***if condition***, if the option is valid or not. A valid option for us would be something ***greater than or equal to*** 1 and ***less than or equal to*** 4. If the option is less valid, we are letting the user know that their input is incorrect with a print statement. If the input is valid, we are taking 2 numbers as input and again converting it to an integer with the ***int()*** function. Next, based on the option user selected, we have a set of ***if, elif and else*** conditions where we are printing the result after performing the relative operation.	
Alright! It seems like now we are getting a hang of Python! In this module, we will be heavily using Python for a lot of cool things we can do with networking, and don't worry if you've forgotten about how Python works. You will be picking up a lot more about Python on the way!	
Teacher Guides Student to Stop Screen Share	
WRAP UP SESSION - 5 Mins	
Can you tell me about what we learnt about in today's class?	**ESR:** We learnt about Python and also got to know about how the ports work in terms of the modern internet, how many ports are there and what ports are accepted widely for which protocols, etc.

Great! In the next class, we will deep dive into what ***sockets*** are and get into ***socket programming*** with Python! It's widely used in creating various sorts of chat applications and bots!	
Quiz time - Click on in-class quiz	
Question	**Answer**
Which data structure in Python has key-value pairs? A. Arrays B. Dictionaries C. Lists D. Objects	**B**
What will be the output of ***list(range(2, 5))***? A. [2, 3, 4, 5] B. [2, 3, 4] C. [0, 1, 2, 3, 4, 5] D. [0, 1, 2, 3, 4]	**B**
What does ***%*** operator do? A. Checks divisibility of one number by another B. Gives quotient of one number divided by another C. Gives percentage of one number out of another D. Gives remainder after one number	**D**

is divided by another	

End the quiz panel

FEEDBACK

- **Appreciate the student for their efforts in the class.**
- **Ask the student to make notes for the reflection journal along with the code they wrote in today's class.**

Teacher Action	Student Action
Did you enjoy today's class? Amazing work today! You get a "hats-off".	**ESR:** Varied. *Make sure you have given at least 2 Hats Off during the class for:*   
Teacher Clicks ✖ End Class	

ADDITIONAL ACTIVITY

Encourage the student to write reflection notes in their reflection journal using markdown. Use these as guiding questions: • What happened today? ○ Describe what happened.	*The student uses the markdown editor to write her/his reflections in the reflection journal*

○ The code I wrote. ● How did I feel after the class? ● What aspects of the class helped me? What did I find difficult?	

Activity	Activity Name	Links
Teacher Activity 1	List of Ports	https://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers
Teacher Activity 2	Google Colab	https://colab.research.google.com/
Teacher Activity 3	Solution Colab	https://colab.research.google.com/drive/1_1mVuBEeF3G0nIqHGVJFLWxd8A5Qnse7?usp=sharing
Student Activity 1	List of Ports	https://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers