



डॉ. ए.पी.जे. अब्दुल कलाम प्रौद्योगिकी संस्थान, टनकपुर, उत्तराखंड

(संघटक संस्थान, उत्तराखंड तकनीकी विश्वविद्यालय, उत्तराखंड सरकार)

Dr. A.P.J. Abdul Kalam Institute of Technology, Tanakpur, Uttarakhand

(A Constituent Institute of Uttarakhand Technical University, Govt. of Uttarakhand)

Approved by AICTE, MHRD New Delhi



LAB FILE

Academic Session (2022-23)

Programme : B. Tech in Computer Science and Engineering
Course : Computer Workshop (Python Programming)
No. of credits : 1
Semester : III
Session : July-Dec 2022
Batch : 2021-2025
Faculty : Dr. Hitesh Kumar Sharma

Submitted By:

Roll No: 211620101001

Name: AAYUSH BIJALWAN

Submitted to:

Dr. Hitesh Kumar Sharma

Dept. of Computer Science

LIST OF EXPERIMENTS

Exp No.	Experiment Name	Date of Experiment
1	How to declare and use variables and operators	
2	Programming using Basic Libraries (Numpy, Pandas, SK Learn etc)	
3	To write a Python program to print HELLO INDIA.	
4	To write a Python program that takes in command line arguments as input and print the number of arguments.	
5	To write a Python program find the division of student.	
6	To write a program implements Fibonacci series.	
7	To write a Python program for factorial.	
8	To write a Python program to use of functions.	
9	To write a Python program to implement list.	
10	To write a Python program to implement tuples.	
11	To write a Python program Insertion sort.	
12	To write a Python program merge sort.	
13	To write a Python program first n prime numbers.	
14	Implementation of Data Science concepts using Python	

LAB EXERCISE 1

Aim:-How to declare and use variables and operators

DESCRIPTION:-Declaring variables

One of the most powerful features of a programming language is the ability to define and manipulate **variables**. A variable is a named location that stores a **value**. Values may be numbers, text, images, sounds, and other types of data. To store a value, you first have to declare a variable.

```
String message;
```

This statement is a **declaration**, because it declares that the variable named `message` has the type `String`. Each variable has a **type** that determines what kind of values it can store. For example, the `int` type can store integers, and the `char` type can store characters.

Some types begin with a capital letter and some with lowercase. We will learn the significance of this distinction later, but for now you should take care to get it right. There is no such type as `Int` or `string`.

To declare an integer variable named `x`, you simply type:

```
int x;
```

Note that `x` is an arbitrary name for the variable. In general, you should use names that indicate what the variables mean. For example, if you saw these declarations, you could probably guess what values would be stored:

```
String firstName;
```

```
String lastName;
```

```
int hour, minute;
```

This example declares two variables with type `String` and two with type `int`. When a variable name contains more than one word, like `firstName`, it is conventional to capitalize the first letter of each word except the first. Variable names are case-sensitive, so `firstName` is not the same as `firstname` or `FirstName`.

This example also demonstrates the **syntax** for declaring multiple variables with the same type on one line: `hour` and `minute` are both integers. Note that each declaration statement ends with a semicolon.

You can use any name you want for a variable. But there are about 50 reserved words, called **keywords**, that you are not allowed to use as variable names. These words include `public`, `class`, `static`, `void`, and `int`, which are used by the compiler to analyze the structure of the program.

2.2 Assignment

Now that we have declared variables, we want to use them to store values. We do that with an **assignment** statement.

```
message = "Hello!"; // give message the value "Hello!"
hour = 11;           // assign the value 11 to hour
minute = 59;         // set minute to 59
```

This example shows three assignments, and the comments illustrate different ways people sometimes talk about assignment statements. The vocabulary can be confusing here, but the idea is straightforward:

- When you declare a variable, you create a named storage location.
- When you make an assignment to a variable, you update its value.

As a general rule, a variable has to have the same type as the value you assign to it. For example, you cannot store a string in `minute` or an integer in `message`. We will see some examples that seem to break this rule, but we'll get to that later.

A common source of confusion is that some strings *look* like integers, but they are not. For example, `message` can contain the string `"123"`, which is made up of the characters `'1'`, `'2'`, and `'3'`. But that is not the same thing as the integer `123`.

```
message = "123"; // legal
```

```
message = 123;           // not legal
```

Variables must be **initialized** (assigned for the first time) before they can be used. You can declare a variable and then assign a value later, as in the previous example. You can also declare and initialize on the same line:

```
String message = "Hello!";  
  
int hour = 11;  
  
int minute = 59;
```

LAB EXERCISE 2

Aim:-Programming using Basic Libraries (Numpy, Pandas, SK Learn etc)

DESCRIPTION:- Till today I have written all tutorials without libraries and now I'm taking our journey to next level where we will use python libraries for classification, visualization and clustering. In this article, we will have a short introduction of NumPy, SciPy, matplotlib, scikit-learn, pandas.

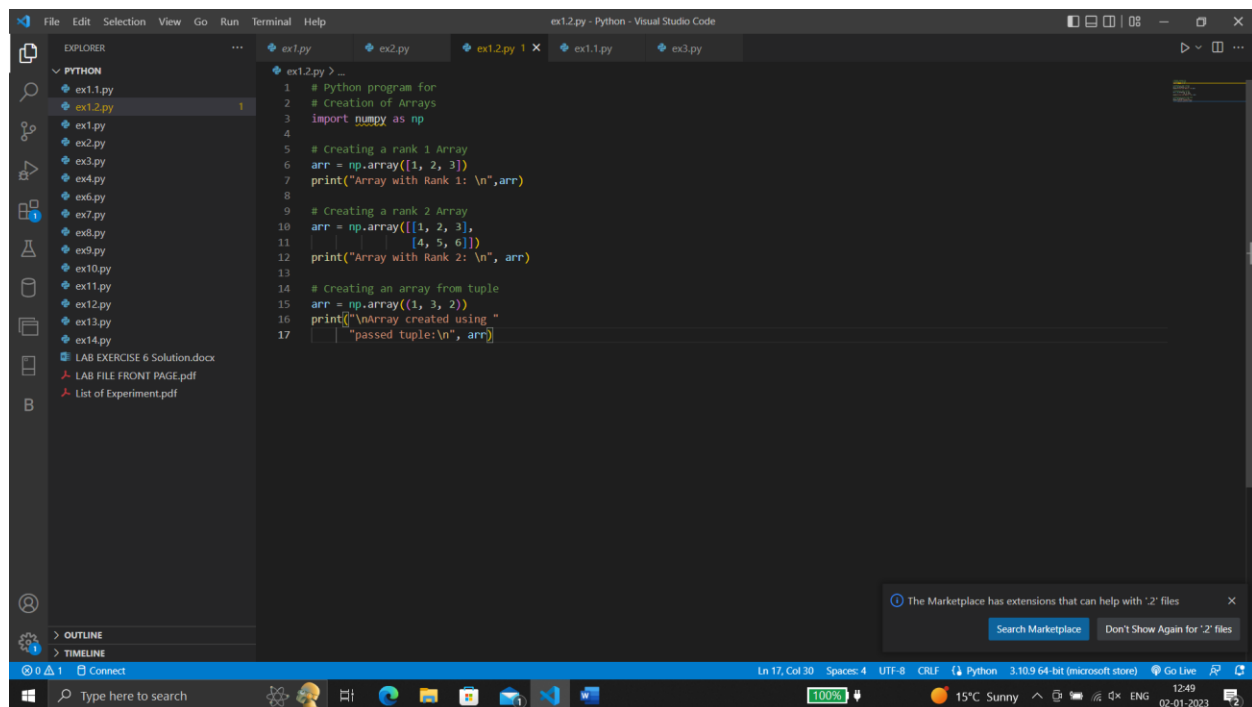
NumPy

NumPy basically provides n-dimensional array object. NumPy also provides mathematical functions which can be used in many calculations.

Command to install: *pip install numpy*

SOLUTION:-

CODE:--



```
1 # Python program for
2 # Creation of Arrays
3 import numpy as np
4
5 # Creating a rank 1 Array
6 arr = np.array([1, 2, 3])
7 print("Array with Rank 1: \n",arr)
8
9 # Creating a rank 2 Array
10 arr = np.array([[1, 2, 3],
11                [4, 5, 6]])
12 print("Array with Rank 2: \n", arr)
13
14 # Creating an array from tuple
15 arr = np.array((1, 3, 2))
16 print("\nArray created using "
17       "passed tuple:\n", arr)
```

OUTPUT:-

```
1 # Python program for
2 # Creation of Arrays
3 import numpy as np
4
5 # Creating a rank 1 Array
6 arr = np.array([1, 2, 3])
7 print("Array with Rank 1: \n",arr)
8
9 # Creating a rank 2 Array
10 arr = np.array([[1, 2, 3],
11                [4, 5, 6]])
12 print("Array with Rank 2: \n", arr)
13
14 # Creating an array from tuple
15 arr = np.array((1, 3, 2))
16 print("\nArray created using "
17       "passed tuple:\n", arr)
```

Array with Rank 1:
[1 2 3]
Array with Rank 2:
[[1 2 3]
 [4 5 6]]
Array created using passed tuple:
[1 3 2]

LAB EXERCISE 3

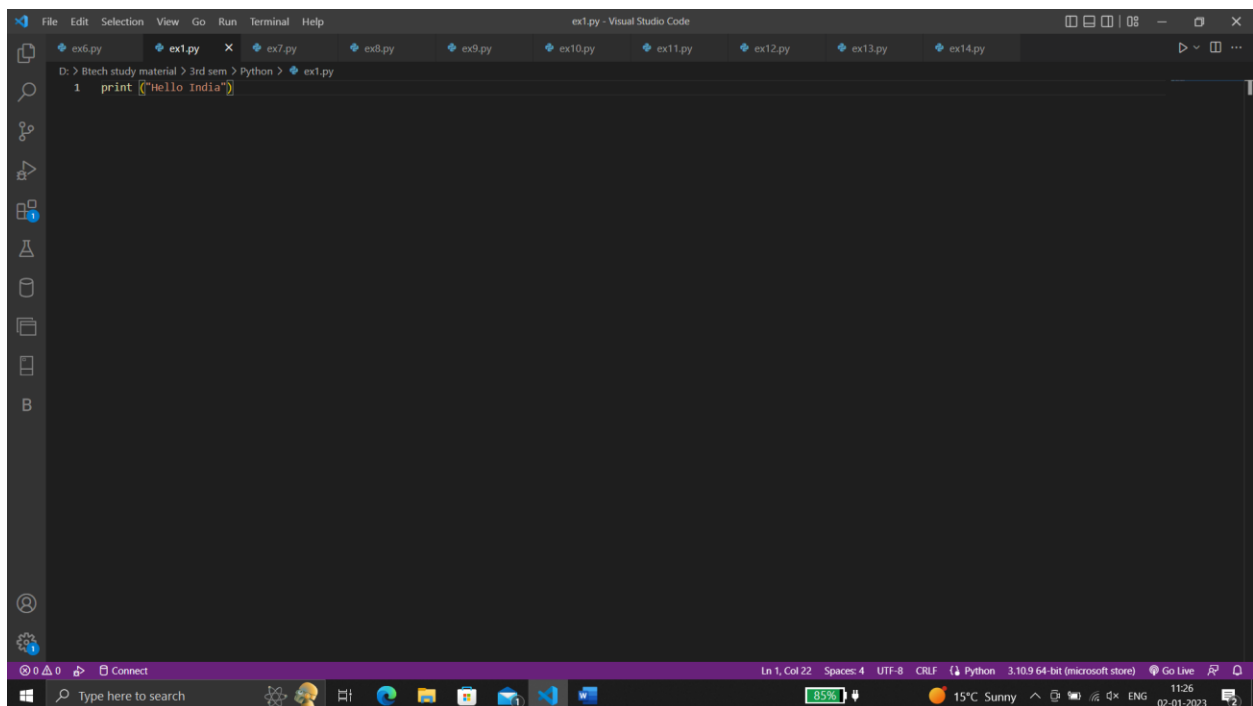
Aim: To write a Python program to print HELLO INDIA.

Description:

In this program, we have used the built-in function to print the string HELLO INDIA on our screen.

Solution:

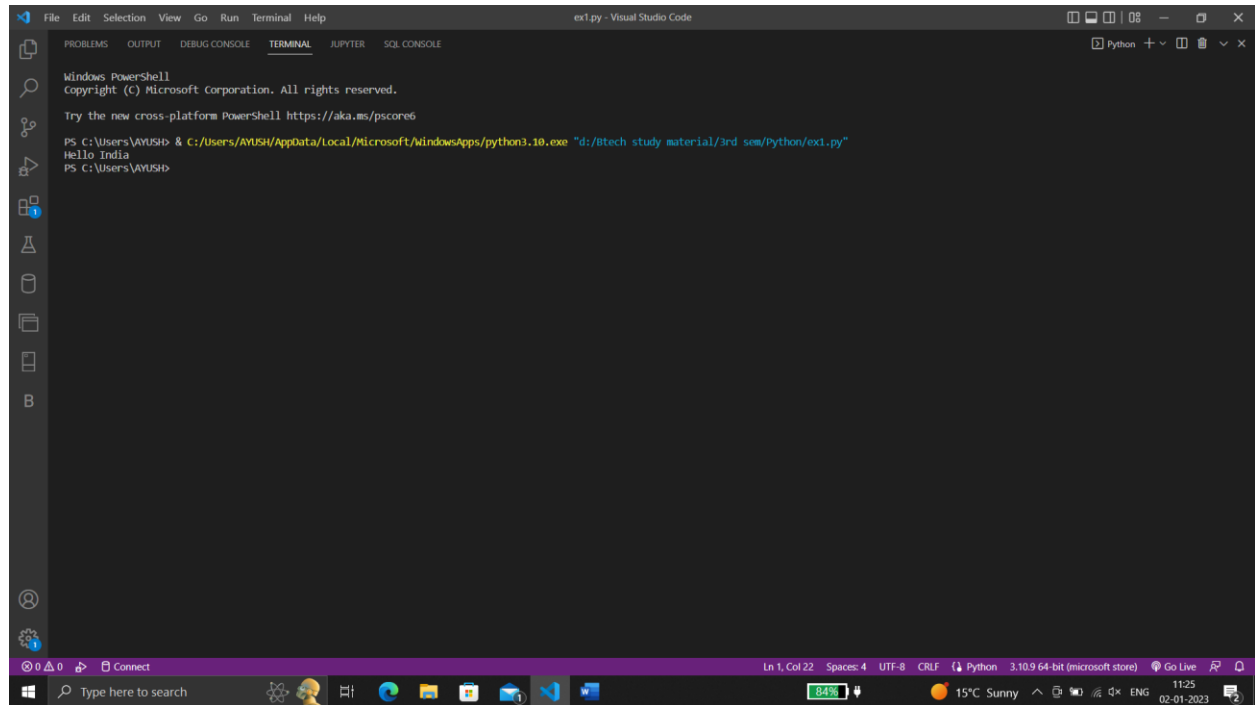
Code:

A screenshot of the Visual Studio Code editor interface. The title bar at the top reads "ex1.py - Visual Studio Code". The menu bar includes File, Edit, Selection, View, Go, Run, Terminal, and Help. The Explorer sidebar on the left shows a file named "ex1.py" selected. The main editor area displays the following code:

```
1 print('Hello India')
```

The status bar at the bottom indicates the current line and column as "Ln 1, Col 22", with 4 spaces, UTF-8 encoding, CRLF line endings, and Python 3.10.9 64-bit (microsoft store). The Windows taskbar at the very bottom shows the search bar, taskbar icons, and system tray information including 85% battery, 15°C Sunny weather, and the date 02-01-2023.

Input/Output:



The screenshot shows a Visual Studio Code window with a terminal open. The terminal is titled "ex1.py - Visual Studio Code" and has tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, JUPYTER, and SQL CONSOLE. The terminal output is as follows:

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\AVUSHD> & C:/Users/AVUSHD/AppData/Local/Microsoft/WindowsApps/python3.10.exe "d:/btech study material/3rd sem/python/ex1.py"
Hello India
PS C:\Users\AVUSHD>
```

The status bar at the bottom of the terminal shows "Ln 1, Col 22", "Spaces: 4", "UTF-8", "CRLF", "Python", "3.10.9 64-bit (microsoft store)", and "Go Live". The Windows taskbar is visible at the bottom of the screen, showing the search bar, taskbar icons, and system tray with a battery level of 84%, temperature of 15°C, and date/time of 11:25 on 02-01-2023.

LAB EXERCISE 4

Aim: To write a Python program that takes in command line arguments as input and print the number of arguments.

Description:

The argument that are given after the name of program in the command line shell of the operating system are known as

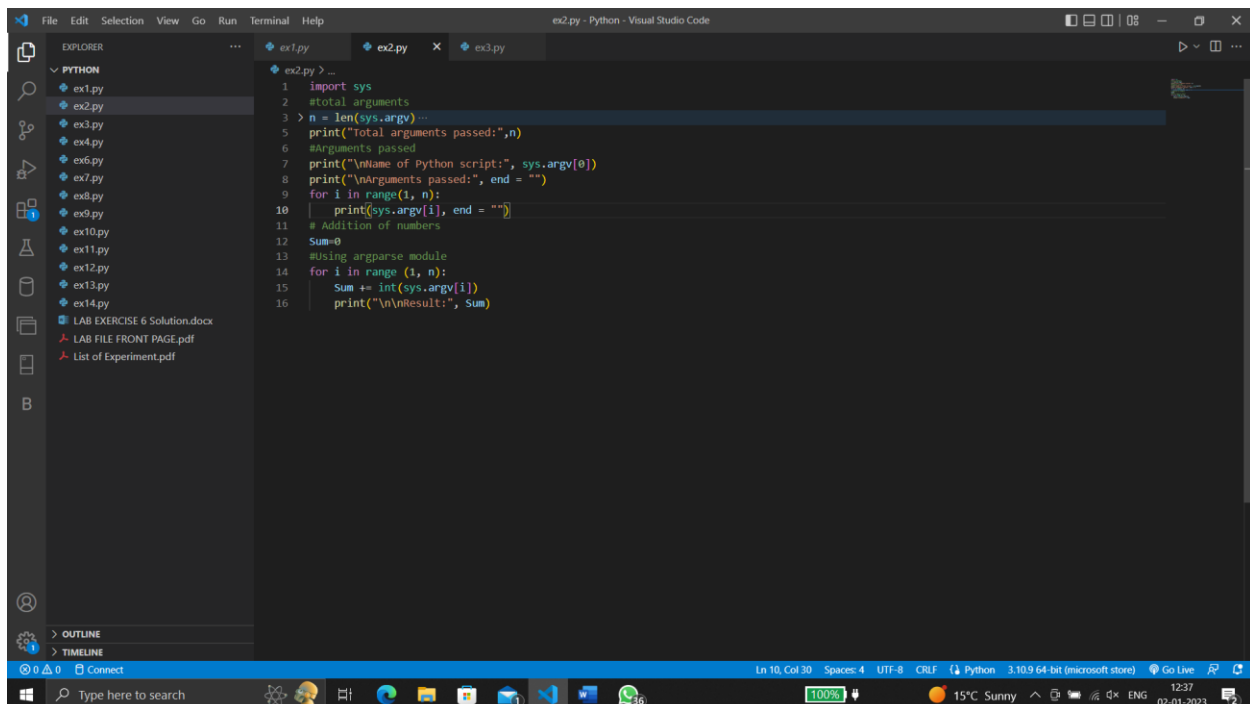
Command Line Arguments.

One such variable is `sys.argv` which is a simple list structure. It's main purpose are:

- It is a list of command line arguments.
- `len(sys.argv)` provides the number of command line arguments.
- `sys.argv[0]` is the name of the current Python script.

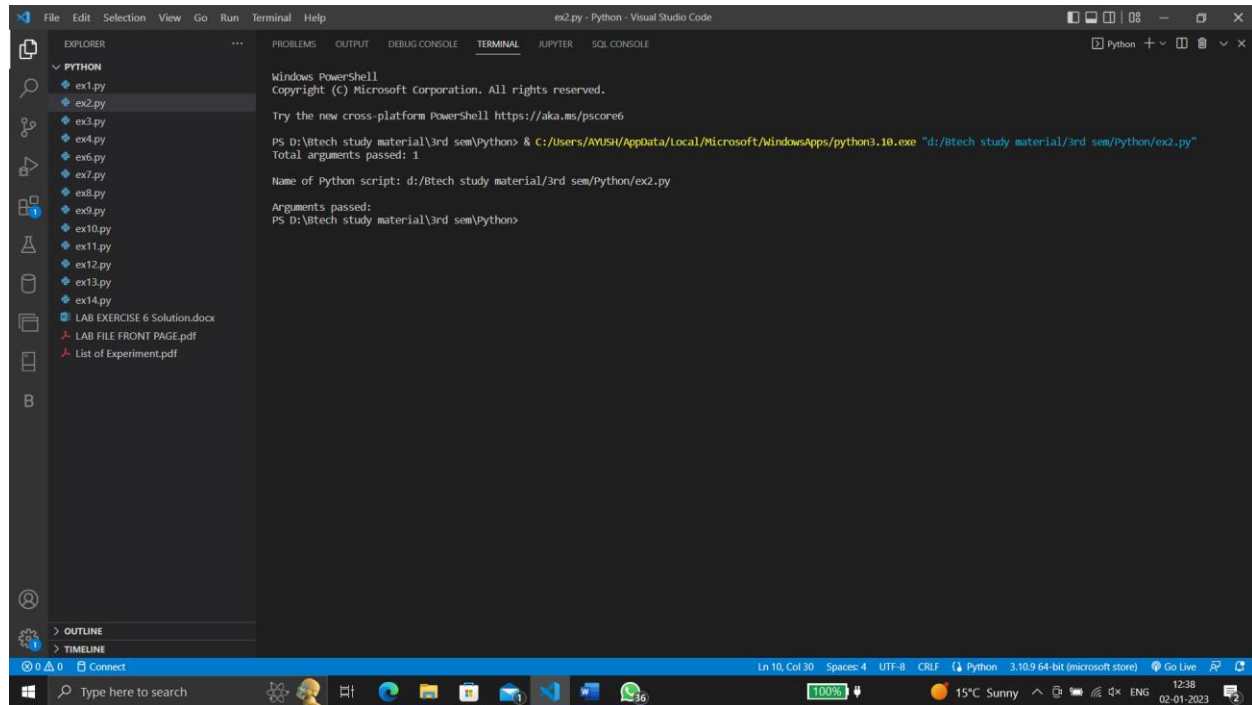
Solution:

Code:



```
1 import sys
2 #total arguments
3 > n = len(sys.argv)
4 print("Total arguments passed:",n)
5 #Arguments passed
6 print("\nName of Python script:", sys.argv[0])
7 print("\nArguments passed:", end = "")
8 for i in range(1, n):
9     print(sys.argv[i], end = " ")
10
11 # Addition of numbers
12 Sum=0
13 #Using argparse module
14 for i in range (1, n):
15     Sum += int(sys.argv[i])
16 print("\n\nResult:", Sum)
```

Input/Output:



The screenshot shows the Visual Studio Code interface with a terminal window open. The terminal displays the output of a PowerShell command executed in a Windows environment. The output includes the PowerShell version, copyright information, a link to the cross-platform PowerShell, the command used to run the Python script, and the arguments passed to it.

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\Btech study material\3rd sem\Python> & C:/Users/AVUSH/AppData/Local/Microsoft/WindowsApps/python3.10.exe "d:/Btech study material/3rd sem/Python/ex2.py"
Total arguments passed: 1

Name of Python script: d:/Btech study material/3rd sem/Python/ex2.py

Arguments passed:
PS D:\Btech study material\3rd sem\Python>
```

The terminal window is titled "ex2.py - Python - Visual Studio Code". The Explorer sidebar on the left shows a list of files in the "PYTHON" folder, including ex1.py through ex14.py, and some PDF files. The bottom status bar shows the current cursor position (Ln 10, Col 30), file encoding (UTF-8), and other details.

LAB EXERCISE 5

Aim: To write a Python program find the division of student.

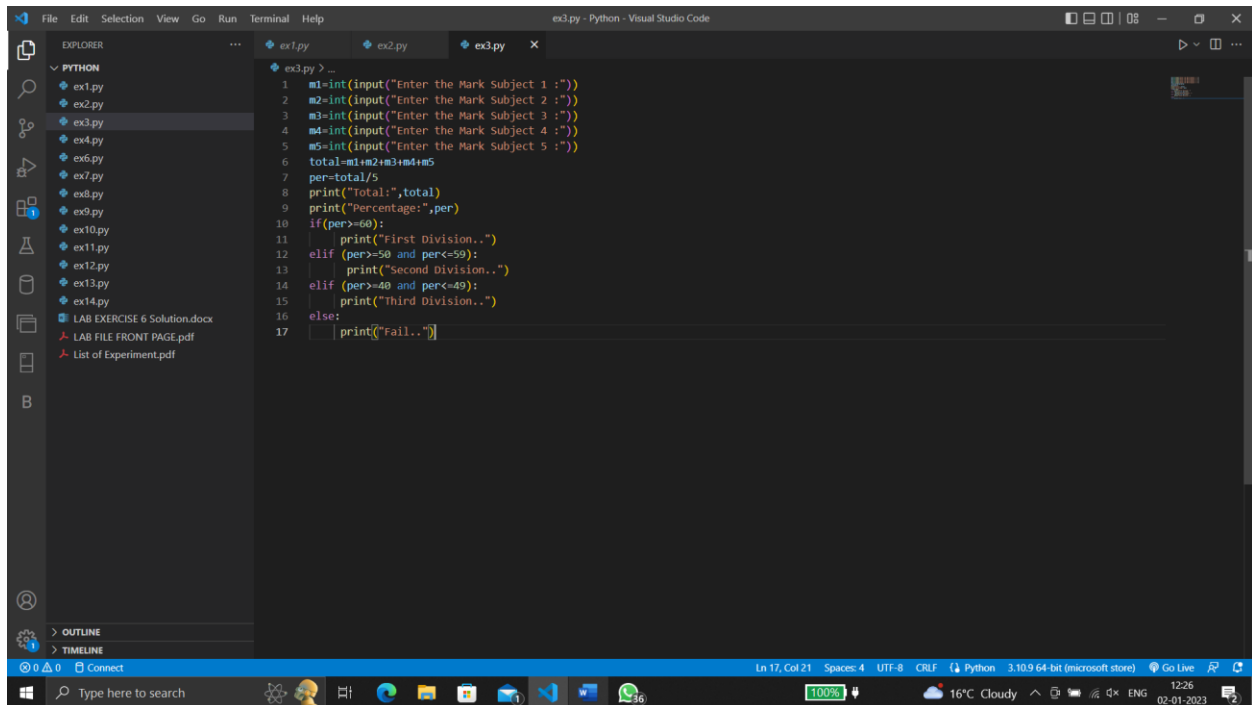
Description:

This program finds and prints the grade of a student based on marks obtained in five subjects entered by the user at run-time.

To calculate the grade of students in Python, you have to ask the user to enter marks obtained in five subjects. Now calculate the sum of all the marks and then calculate the average marks to find the grade according to the average marks obtained by the students.

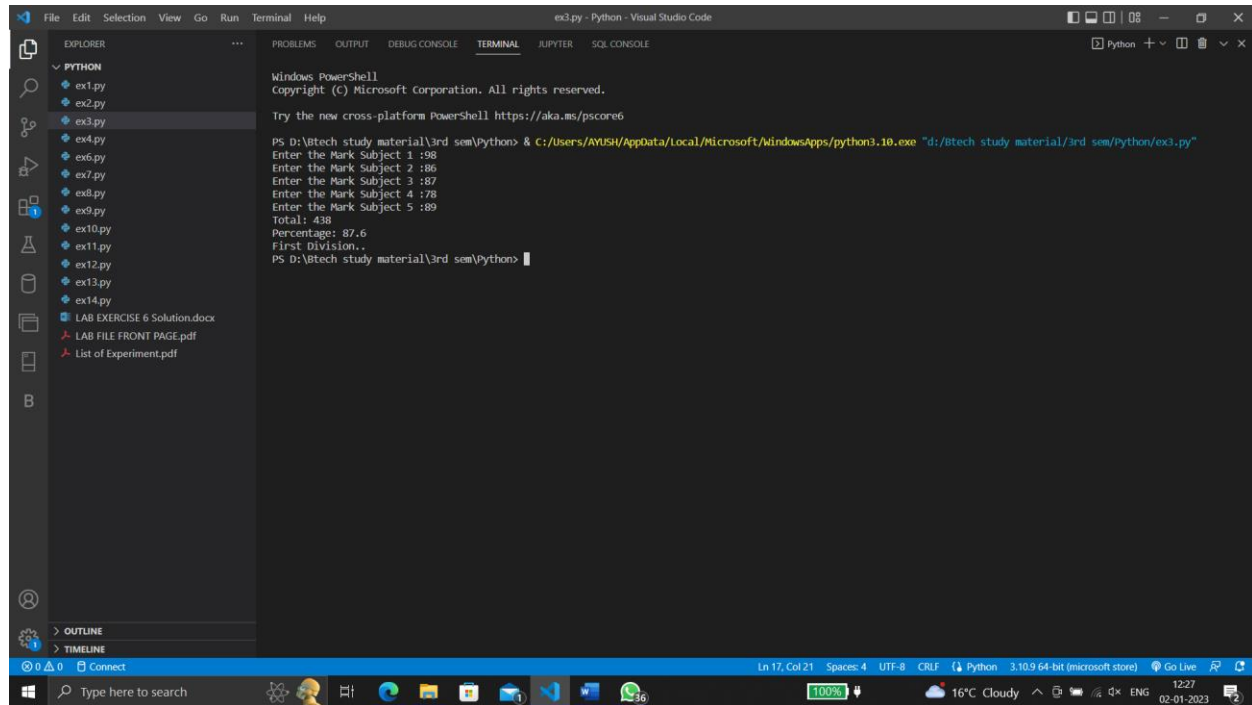
Solution:

Code:

The image shows a screenshot of the Visual Studio Code editor interface. The Explorer pane on the left shows a file tree with a 'PYTHON' folder containing files 'ex1.py' through 'ex14.py', and a 'LAB EXERCISE 6' folder containing 'LAB EXERCISE 6 Solution.docx', 'LAB FILE FRONT PAGE.pdf', and 'List of Experiment.pdf'. The main editor window displays the code for 'ex3.py'. The code prompts the user to enter marks for five subjects, calculates the total and percentage, and then prints the division based on the percentage. The status bar at the bottom indicates the current line and column (Ln 17, Col 21), the number of spaces (4), the encoding (UTF-8), the line ending (CRLF), the Python version (3.10.9 64-bit), and the Go Live extension. The Windows taskbar at the very bottom shows the search bar, task view, and several open applications including a web browser, file explorer, and a chat application. The system tray shows the date and time as 12:26 on 02-01-2023, along with weather information (16°C Cloudy) and network status (100%).

```
ex3.py - Python - Visual Studio Code
1  m1=int(input("Enter the Mark Subject 1 :"))
2  m2=int(input("Enter the Mark Subject 2 :"))
3  m3=int(input("Enter the Mark Subject 3 :"))
4  m4=int(input("Enter the Mark Subject 4 :"))
5  m5=int(input("Enter the Mark Subject 5 :"))
6  total=m1+m2+m3+m4+m5
7  per=total/5
8  print("Total:",total)
9  print("Percentage:",per)
10 if(per>=60):
11     print("First Division..")
12 elif (per>=50 and per<=59):
13     print("Second Division..")
14 elif (per>=40 and per<=49):
15     print("Third Division..")
16 else:
17     print("Fail..")
```

Input/Output:



The screenshot shows the Visual Studio Code interface with a terminal window open. The terminal displays the output of a Python script executed in a Windows PowerShell environment. The script prompts the user to enter marks for five subjects, calculates the total, percentage, and first division.

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\Btech study material\3rd sem\Python> & c:/Users/AVUSH/AppData/Local/Microsoft/WindowsApps/python3.10.exe "d:/Btech study material/3rd sem/python/ex3.py"
Enter the Mark Subject 1 :98
Enter the Mark Subject 2 :86
Enter the Mark Subject 3 :87
Enter the Mark Subject 4 :78
Enter the Mark Subject 5 :89
Total: 438
Percentage: 87.6
First Division..
PS D:\Btech study material\3rd sem\Python>
```

The Explorer sidebar on the left shows a project structure with a 'PYTHON' folder containing files 'ex1.py' through 'ex14.py', and a 'LAB EXERCISE 6 Solution.docx' file. The bottom status bar indicates the current file is 'ex3.py' at line 17, column 21, with a UTF-8 encoding and CRLF line endings.

LAB EXERCISE 6

Aim: To write a program implements Fibonacci series.

Description:

The Fibonacci numbers are the numbers in the following integer sequence.
0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144,

In mathematical terms, the sequence F_n of Fibonacci numbers is defined by the recurrence relation

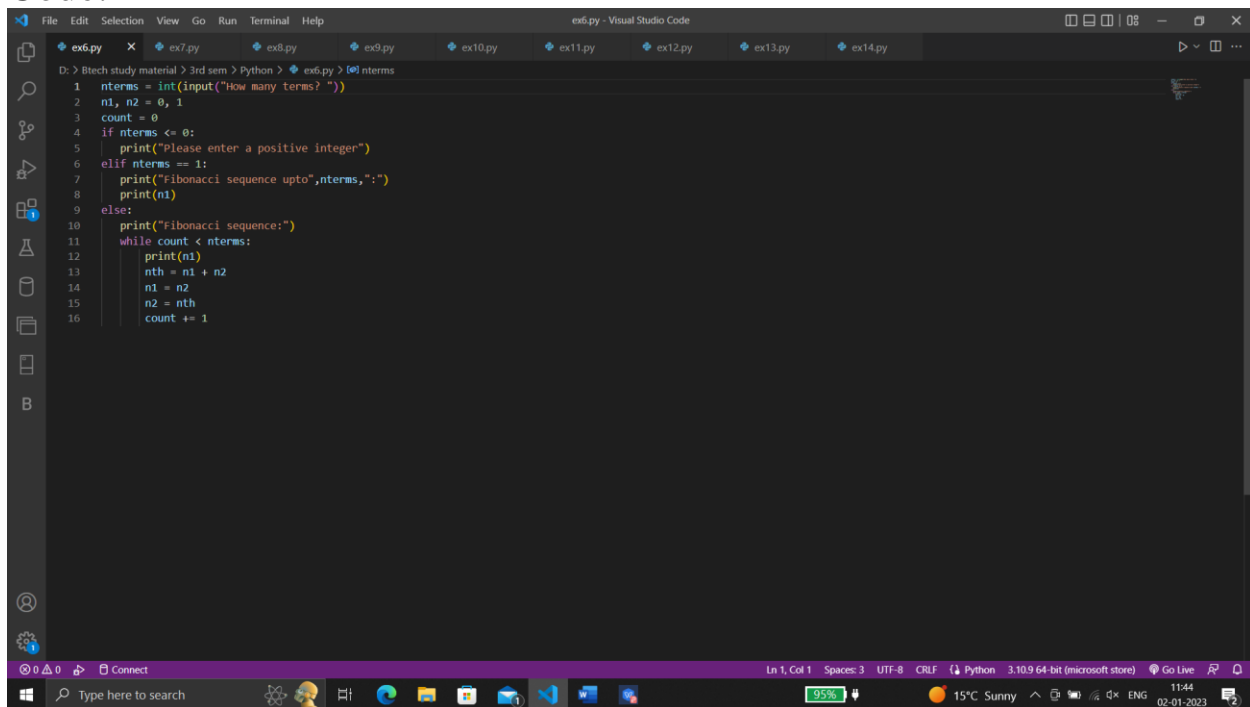
$$F_n = F_{n-1} + F_{n-2}$$

with seed values

$$F_0 = 0 \text{ and } F_1 = 1.$$

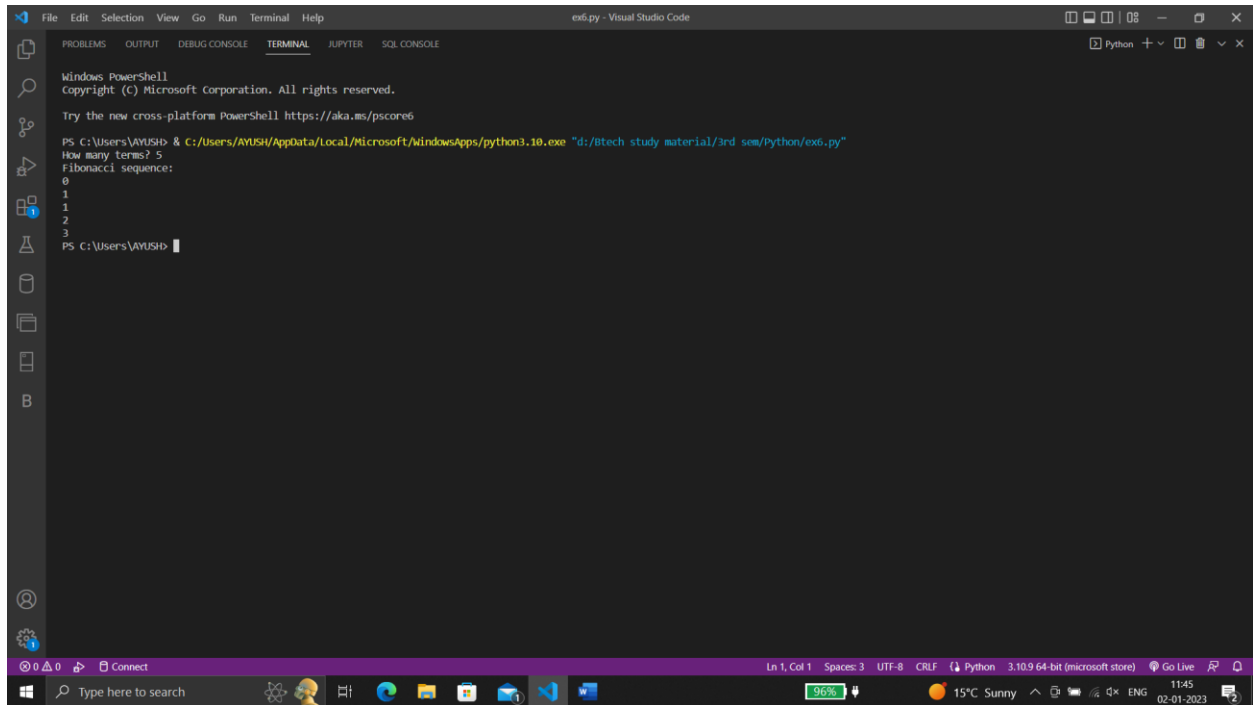
Solution:

Code:



```
File Edit Selection View Go Run Terminal Help
ex6.py - Visual Studio Code
ex6.py ex7.py ex8.py ex9.py ex10.py ex11.py ex12.py ex13.py ex14.py
D:\Btech study material > 3rd sem > Python > ex6.py [0] nterms
1 nterms = int(input("How many terms? "))
2 n1, n2 = 0, 1
3 count = 0
4 if nterms <= 0:
5     print("Please enter a positive integer")
6 elif nterms == 1:
7     print("Fibonacci sequence upto",nterms,":")
8     print(n1)
9 else:
10    print("Fibonacci sequence:")
11    while count < nterms:
12        print(n1)
13        nth = n1 + n2
14        n1 = n2
15        n2 = nth
16        count += 1
```

Input/Output:



```
File Edit Selection View Go Run Terminal Help
ex6.py - Visual Studio Code
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER SQL CONSOLE
Python + - + + + +
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\AVUSHB & C:/Users/AVUSHB/AppData/Local/Microsoft/WindowsApps/python3.10.exe "d:/btech study material/3rd sem/python/ex6.py"
How many terms? 5
Fibonacci sequence:
0
1
1
2
3
PS C:\Users\AVUSHB
```

Ln 1, Col 1 Spaces: 3 UTF-8 CRLF Python 3.10.9 64-bit (microsoft store) Go Live 96% 15°C Sunny 11:45 02-01-2023

LAB EXERCISE 7

Aim: To write a Python program for factorial

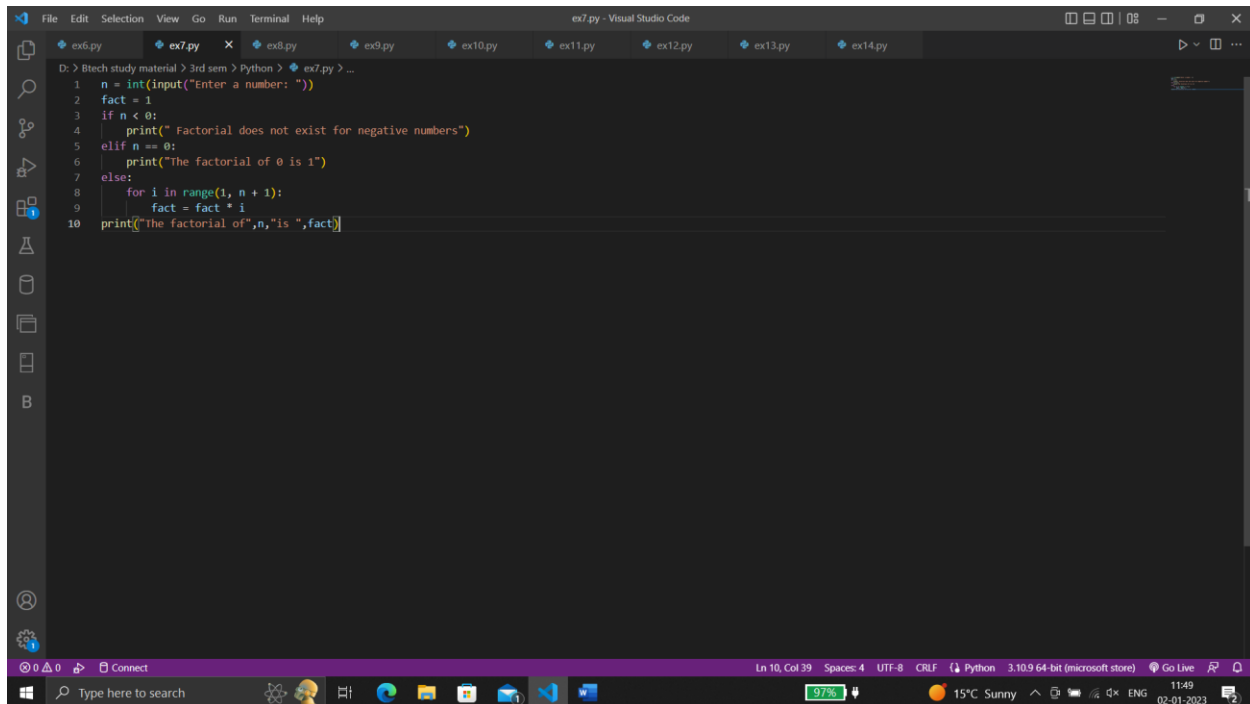
Description:

factorial is a simple thing. Factorials are just products. An exclamation mark indicates the factorial. Factorial is a multiplication operation of natural numbers with all the natural number that are less than it.

$$n! = n \times (n-1) \times (n-2) \times (n-3) \times \dots \times 3 \times 2 \times 1$$

Solution:

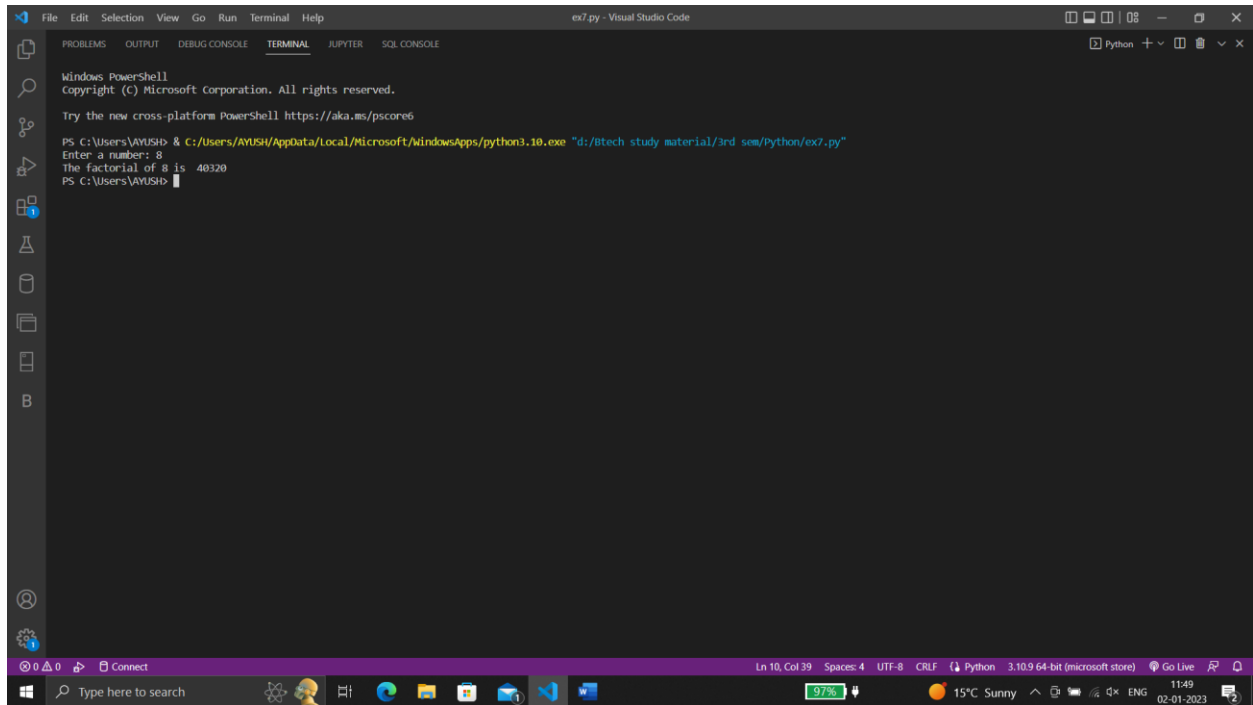
Code:



```
File Edit Selection View Go Run Terminal Help
ex7.py - Visual Studio Code
ex6.py ex7.py x ex8.py ex9.py ex10.py ex11.py ex12.py ex13.py ex14.py
D:\Btech study material > 3rd sem > Python > ex7.py > ...
1 n = int(input("Enter a number: "))
2 fact = 1
3 if n < 0:
4     print("Factorial does not exist for negative numbers")
5 elif n == 0:
6     print("The factorial of 0 is 1")
7 else:
8     for i in range(1, n + 1):
9         fact = fact * i
10 print("The factorial of",n,"is ",fact)
```

The screenshot shows the Visual Studio Code interface with a Python file named ex7.py. The code calculates the factorial of a user-input number. It includes comments for negative numbers and zero. The output shows the factorial of 5 is 120.

Input/Output:



The screenshot shows a Visual Studio Code window with a terminal open. The terminal is running a PowerShell session. The output of the session is as follows:

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\AVUSHB & C:\Users\AVUSHB\AppData\Local\Microsoft\WindowsApps\python3.10.exe "d:/btech study material/3rd sem/python/ex7.py"
Enter a number: 8
The factorial of 8 is 40320
PS C:\Users\AVUSHB
```

The terminal window is titled "ex7.py - Visual Studio Code". The status bar at the bottom shows the current line and column as "Ln 10, Col 39", the file encoding as "UTF-8", the line ending as "CRLF", the Python version as "Python 3.10.9 64-bit (microsoft store)", and the Go Live status as "Go Live". The system tray at the bottom right shows the date and time as "11:49 02-01-2023".

LAB EXERCISE 8

Aim: To write a Python program to use of functions

Description:

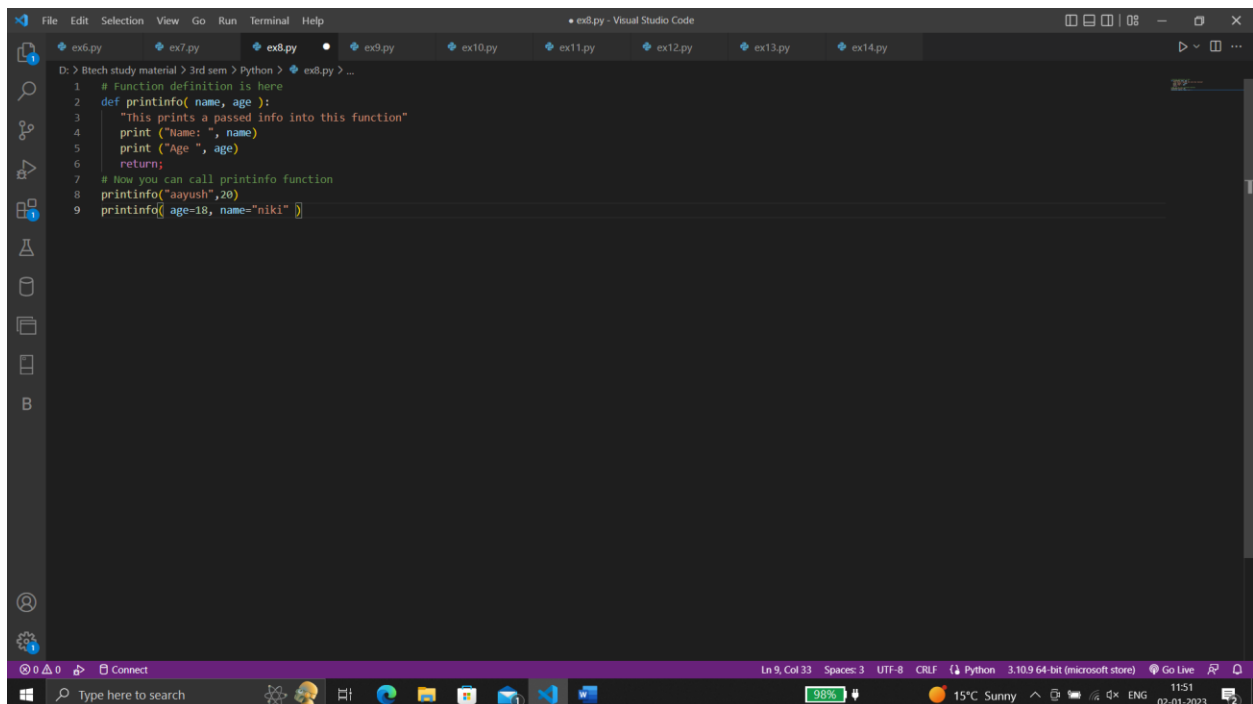
A function is a block of code which only runs when it is called.

You can pass data, known as parameters, into a function.

A function can return data as a result.

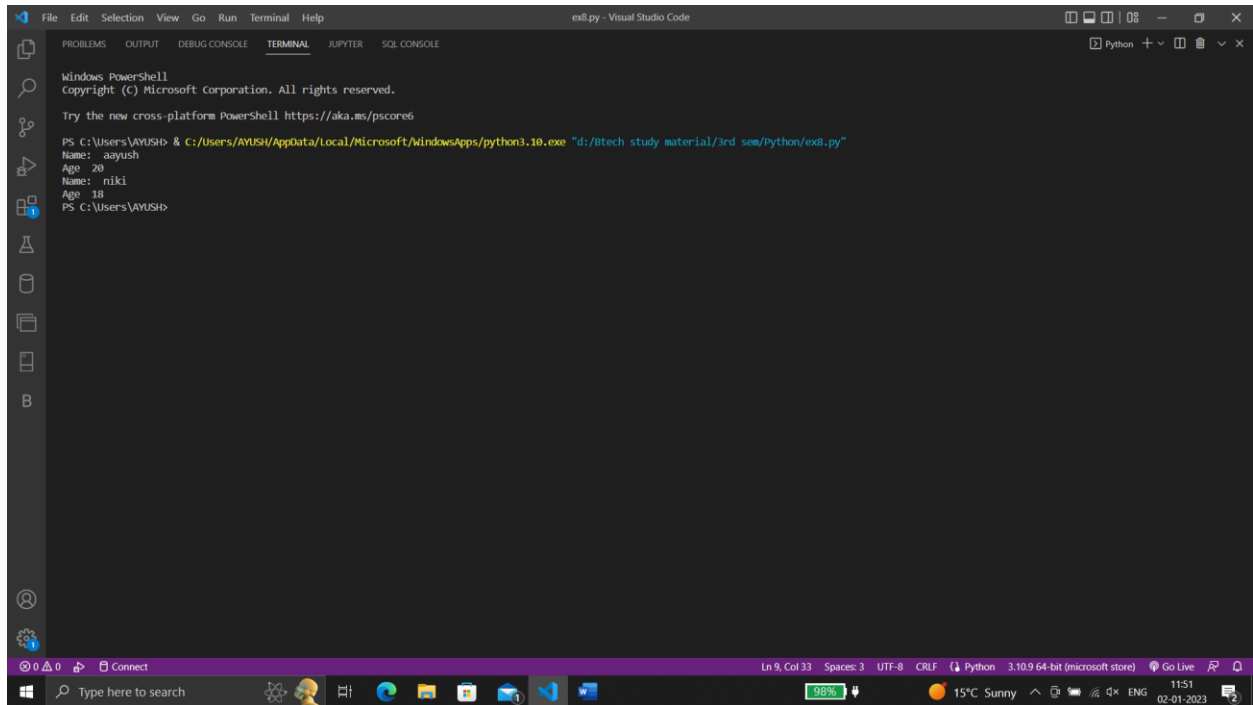
Solution:

Code:



```
1 # Function definition is here
2 def printinfo( name, age ):
3     "This prints a passed info into this function"
4     print ("Name: ", name)
5     print ("Age ", age)
6     return;
7
8 # Now you can call printinfo function
9 printinfo("aayush",20)
10 printinfo( age=18, name="niki" )
```

Input/Output:



The image shows a screenshot of a Visual Studio Code window with a terminal open. The terminal is titled "ex8.py - Visual Studio Code" and shows the output of a PowerShell command. The output includes the Windows PowerShell copyright notice, a link to the new cross-platform PowerShell, and the execution of a Python script. The script outputs the names and ages of two people: aayush (20) and niki (18).

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\AVUSH> & C:/Users/AVUSH/AppData/Local/Microsoft/WindowsApps/python3.10.exe "d:/btech study material/3rd sem/python/ex8.py"
Name: aayush
Age: 20
Name: niki
Age: 18
PS C:\Users\AVUSH>
```

The Visual Studio Code interface shows the "TERMINAL" tab selected. The status bar at the bottom indicates the current file is "ex8.py" at line 9, column 33, with 3 spaces, UTF-8 encoding, and CRLF line endings. The Python version is 3.10.9 64-bit (microsoft store). The system tray shows a battery level of 98%, a temperature of 15°C, and the date 02-01-2023.

LAB EXERCISE 9

Aim: To write a Python program to implement list

Description:

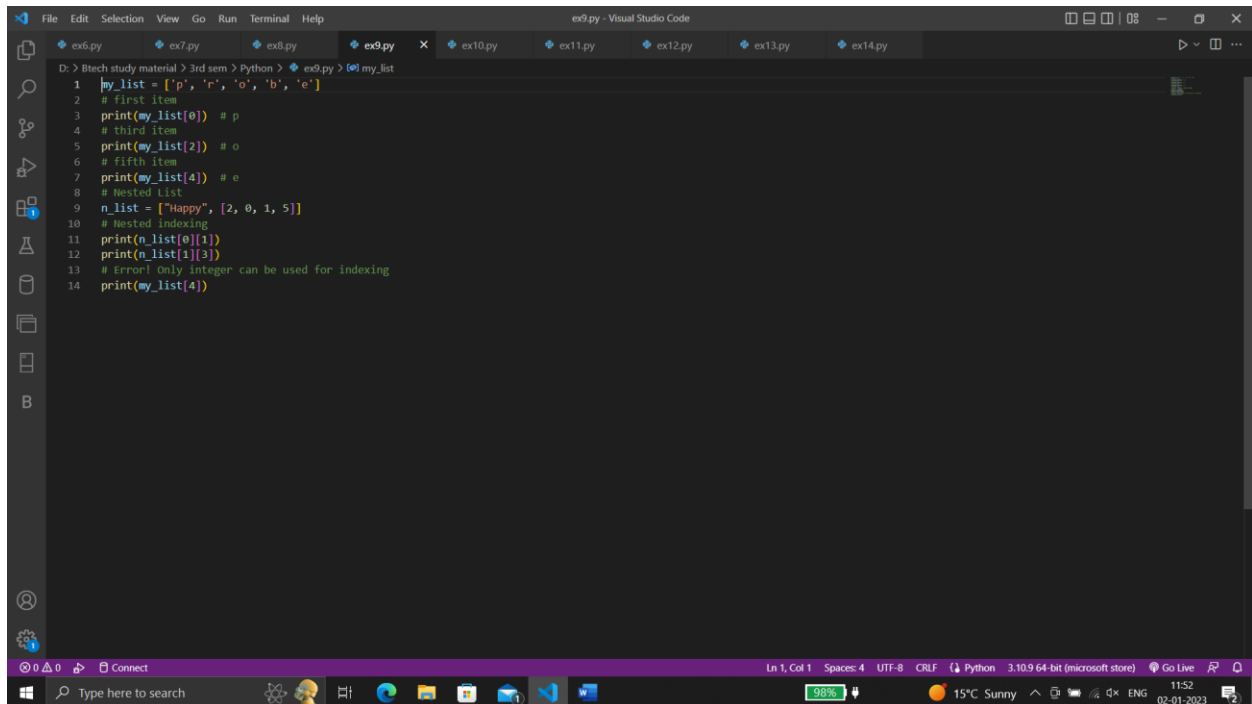
Lists are used to store multiple items in a single variable.

Lists are one of 4 built-in data types in Python used to store collections of data, the other 3 are Tuple, Set, and Dictionary, all with different qualities and usage.

Lists are created using square brackets:

Solution:

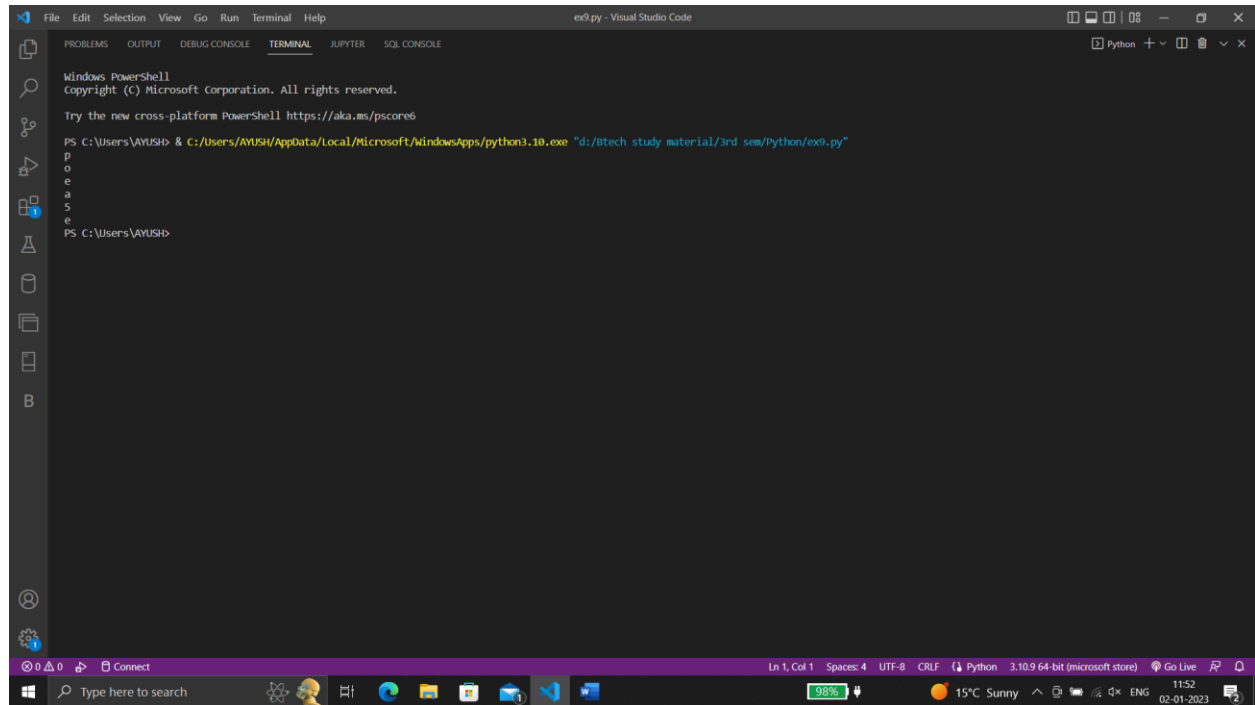
Code:

A screenshot of the Visual Studio Code editor interface. The editor window shows a Python file named 'ex9.py' with the following code:

```
1 my_list = ['p', 'r', 'o', 'b', 'e']
2 # first item
3 print(my_list[0]) # p
4 # third item
5 print(my_list[2]) # o
6 # fifth item
7 print(my_list[4]) # e
8 # nested list
9 n_list = ["Happy", [2, 0, 1, 5]]
10 # nested indexing
11 print(n_list[0][1])
12 print(n_list[1][3])
13 # Error! only integer can be used for indexing
14 print(my_list[4])
```

The interface includes a sidebar on the left with icons for Explorer, Search, Source Control, and Run and Debug. The bottom status bar shows 'Ln 1, Col 1', 'Spaces: 4', 'UTF-8', 'CRLF', 'Python', '3.10.9 64-bit (microsoft store)', 'Go Live', and a battery level of 98%. The Windows taskbar is visible at the very bottom with the search bar and several application icons.

Input/Output:



The image shows a screenshot of a Visual Studio Code terminal window. The terminal is titled "ex9.py - Visual Studio Code". The menu bar includes File, Edit, Selection, View, Go, Run, Terminal, and Help. The terminal tabs are PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, JUPYTER, and SQL CONSOLE. The terminal content shows a Windows PowerShell prompt. The first line is "Windows PowerShell", followed by "Copyright (C) Microsoft Corporation. All rights reserved." and "Try the new cross-platform PowerShell https://aka.ms/pscore6". The second line is a command: "PS C:\Users\AVUSHB & C:/Users/AVUSH/AppData/Local/Microsoft/WindowsApps/python3.10.exe "d:/btech study material/3rd sem/python/ex9.py"". The output shows the command being executed, with the prompt changing to "PS C:\Users\AVUSHB>". The status bar at the bottom shows "Ln 1, Col 1", "Spaces: 4", "UTF-8", "CRLF", "Python", "3.10.9 64-bit (microsoft store)", "Go Live", and a battery level of 98%.

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\AVUSHB & C:/Users/AVUSH/AppData/Local/Microsoft/WindowsApps/python3.10.exe "d:/btech study material/3rd sem/python/ex9.py"
p
o
e
a
s
e
PS C:\Users\AVUSHB>
```

LAB EXERCISE 10

Aim: To write a Python program to implement tuples.

Description:

Tuples are used to store multiple items in a single variable.

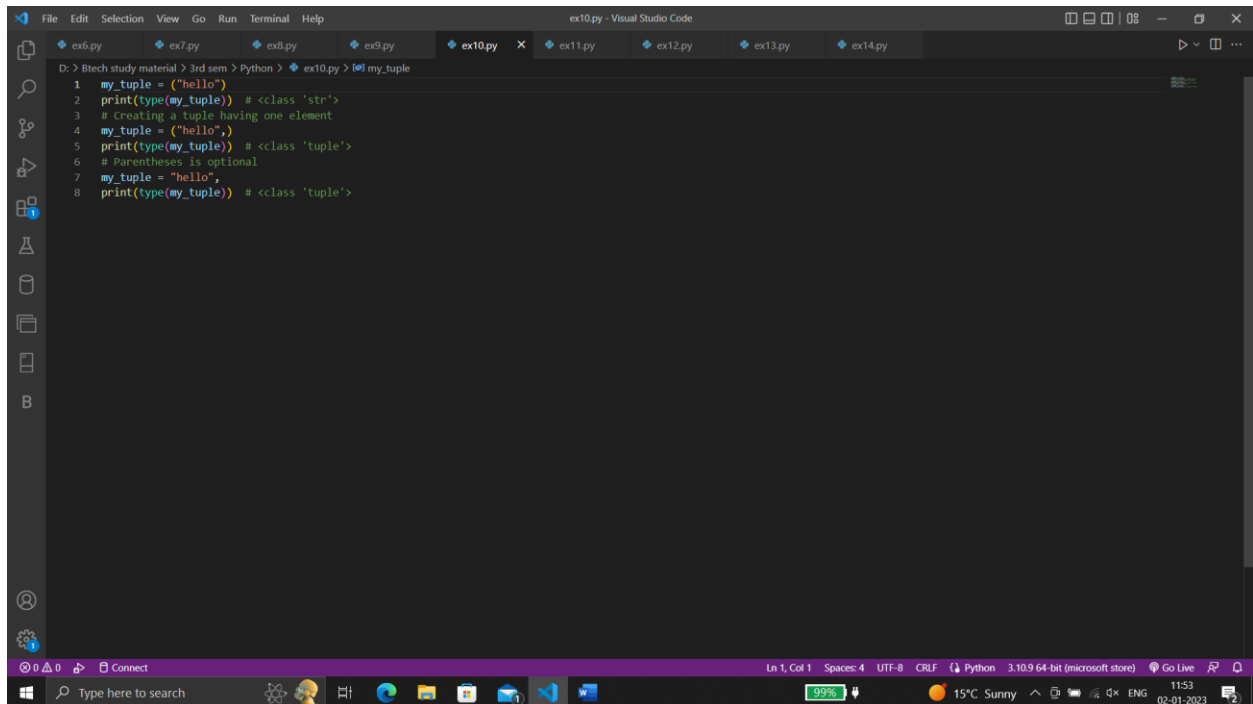
Tuple is one of 4 built-in data types in Python used to store collections of data, the other 3 are List, Set, and Dictionary, all with different qualities and usage.

A tuple is a collection which is ordered and unchangeable.

Tuples are written with round brackets.

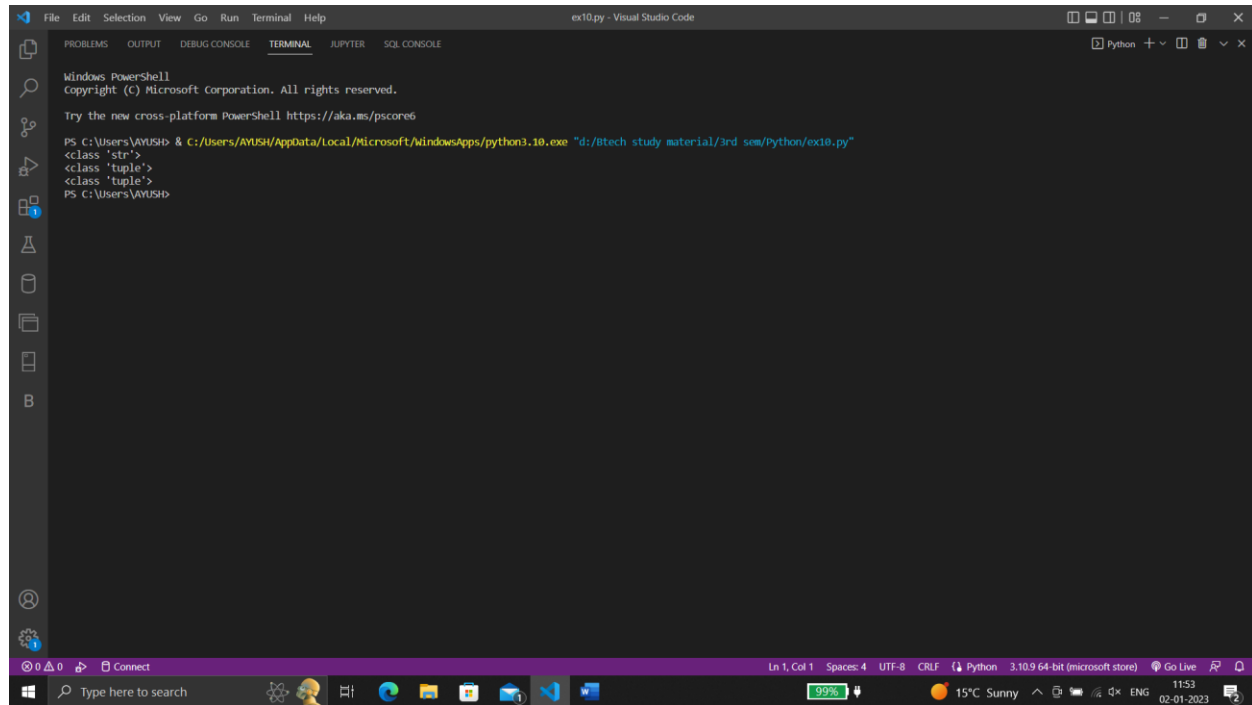
Solution:

Code:



```
1 my_tuple = ("hello")
2 print(type(my_tuple)) # <class 'str'>
3 # Creating a tuple having one element
4 my_tuple = ("hello",)
5 print(type(my_tuple)) # <class 'tuple'>
6 # Parentheses is optional
7 my_tuple = "hello",
8 print(type(my_tuple)) # <class 'tuple'>
```

Input/Output:



The image shows a screenshot of a Visual Studio Code window with a terminal open. The terminal is titled "ex10.py - Visual Studio Code" and shows the output of a PowerShell command. The output includes the Windows PowerShell copyright notice, a link to the new cross-platform PowerShell, and the execution of a Python script. The script prints the class types of 'str' and 'tuple' objects.

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\AVUSH> & C:/Users/AVUSH/AppData/Local/Microsoft/WindowsApps/python3.10.exe "d:/btech study material/3rd sem/python/ex10.py"
<class 'str'>
<class 'tuple'>
<class 'tuple'>
PS C:\Users\AVUSH>
```

The bottom status bar of the terminal shows "Ln 1, Col 1", "Spaces: 4", "UTF-8", "CRLF", "Python", "3.10.9 64-bit (microsoft store)", and "Go Live". The Windows taskbar at the bottom shows the search bar, taskbar icons, system tray with 99% battery, 15°C Sunny, and the date/time 11:53 on 02-01-2023.

LAB EXERCISE 11

Aim: To write a Python program Insertion sort.

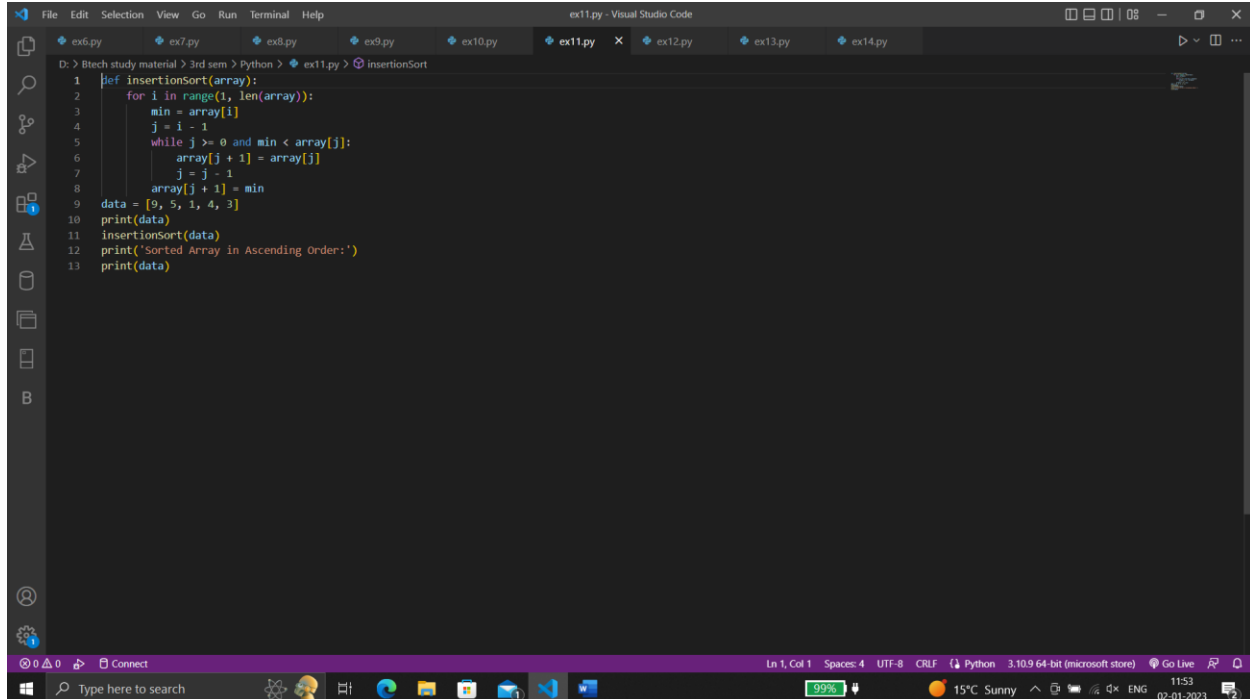
Description:

Insertion sort is the simple method of sorting an array. In this technique, the array is virtually split into the sorted and unsorted part. An element from unsorted part is picked and is placed at correct position in the sorted part.

- The array elements are traversed from 1 to n.
- If the array element at position i is greater than its predecessor, it does not need to be moved.

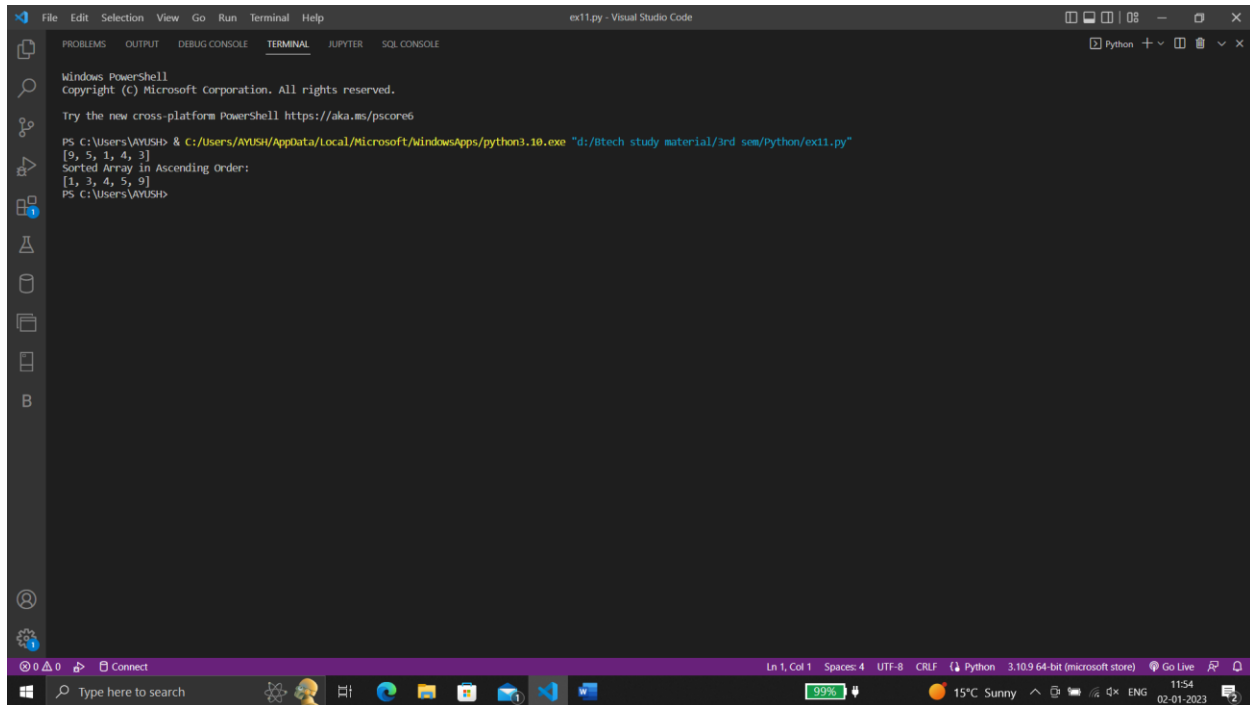
Solution:

Code:



```
1 def insertionSort(array):
2     for i in range(1, len(array)):
3         min = array[i]
4         j = i - 1
5         while j >= 0 and min < array[j]:
6             array[j + 1] = array[j]
7             j = j - 1
8         array[j + 1] = min
9 data = [9, 5, 1, 4, 3]
10 print(data)
11 insertionSort(data)
12 print('Sorted Array in Ascending Order:')
13 print(data)
```


Input/Output:



The screenshot shows a Visual Studio Code window with a terminal open. The terminal is running a Windows PowerShell session. The output of the command `python3.10.exe "d:/btech study material/3rd sem/python/ex1.py"` is displayed. The output shows the array `[9, 5, 1, 4, 3]` and its sorted version `[1, 3, 4, 5, 9]`.

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\AVUSHB & C:/Users/AVUSHB/AppData/Local/Microsoft/WindowsApps/python3.10.exe "d:/btech study material/3rd sem/python/ex1.py"
[9, 5, 1, 4, 3]
Sorted Array in Ascending Order:
[1, 3, 4, 5, 9]
PS C:\Users\AVUSHB
```

LAB EXERCISE 12

Aim: To write a Python program merge sort.

Description:

Merge Sort is a Divide and Conquer algorithm. It divides input array in two halves, calls itself for the two halves and then merges the two sorted halves. The `merge()` function is used for merging two halves.

The `merge(arr, l, m, r)` is key process that assumes that `arr[l..m]` and `arr[m+1..r]` are sorted and merges the two sorted sub-arrays into one.

The sub lists are divided again and again into halves until we get the only one element each. Then we combine the pair of one element lists into two element lists, sorting them in the process.

The sorted two element pairs is merged into the four element lists, and so on until we get the sorted list.

Solution:

Code:

The screenshot shows the Visual Studio Code editor with a Python file named `ex12.py` open. The code implements a merge sort algorithm. It includes a `merge` function that takes an array and two indices `l` and `r`, and a `mergeSort` function that recursively sorts the array. The array to be sorted is `[12, 11, 13, 5, 6, 7]`. The code prints the original array and the sorted array. The output of the program is visible in the terminal window at the bottom of the editor.

```
1 def merge(arr, l, m, r):
2     n1 = m - l + 1
3     n2 = r - m
4     L = [0] * (n1)
5     R = [0] * (n2)
6     for i in range(0, n1):
7         L[i] = arr[l + i]
8     for j in range(0, n2):
9         R[j] = arr[m + 1 + j]
10    i = 0
11    j = 0
12    k = l
13    while i < n1 and j < n2:
14        if L[i] <= R[j]:
15            arr[k] = L[i]
16            i += 1
17        else:
18            arr[k] = R[j]
19            j += 1
20        k += 1
21    while i < n1:
22        arr[k] = L[i]
23        i += 1
24        k += 1
25    while j < n2:
26        arr[k] = R[j]
27        j += 1
28        k += 1
29    def mergeSort(arr, l, r):
30        if l < r:
31            m = l + (r - l) // 2
32            mergeSort(arr, l, m)
33            mergeSort(arr, m + 1, r)
34            merge(arr, l, m, r)
35    arr = [12, 11, 13, 5, 6, 7]
36    n = len(arr)
37    print("Given array is")
38    for i in range(n):
39        print("%d" % arr[i], end=" ")
40    mergeSort(arr, 0, n - 1)
41    print("\n\nSorted array is")
42    for i in range(n):
43        print("%d" % arr[i], end=" ")
44
```

Input/Output:

The screenshot shows the Visual Studio Code terminal window with the output of the Python program. The output displays the original array `12 11 13 5 6 7` and the sorted array `5 6 7 11 12 13`.

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\AVUSH> & C:\Users\AVUSH\AppData\Local\Microsoft\WindowsApps\python3.10.exe "d:/Btech study material/3rd sem/python/ex12.py"
Given array is
12 11 13 5 6 7

Sorted array is
5 6 7 11 12 13
PS C:\Users\AVUSH>
```

LAB EXERCISE 13

Aim: To write a Python program first n prime numbers

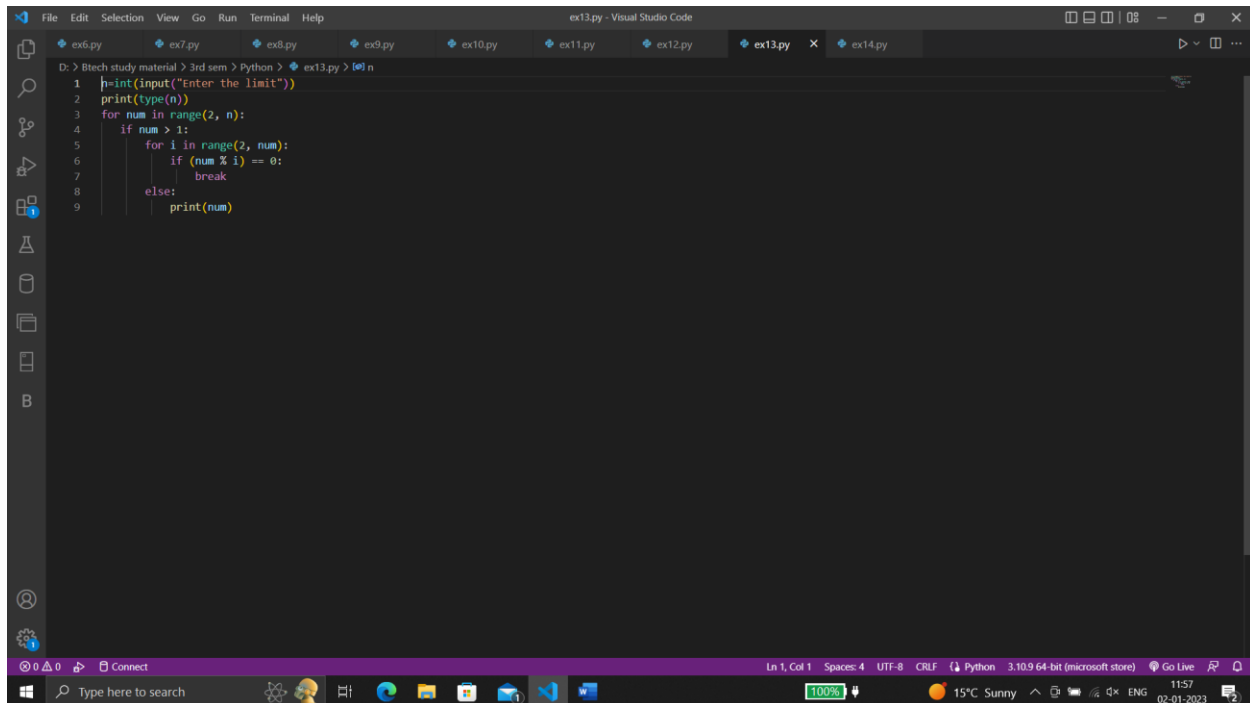
Description:

A prime number is a natural number which is greater than 1 and has no positive divisor other than 1 and itself, such as 2, 3, 5, 7, 11, 13, and so on.

The user is given two integer numbers, lower value, and upper value. The task is to write the Python program for printing all the prime numbers between the given interval (or range).

Solution:

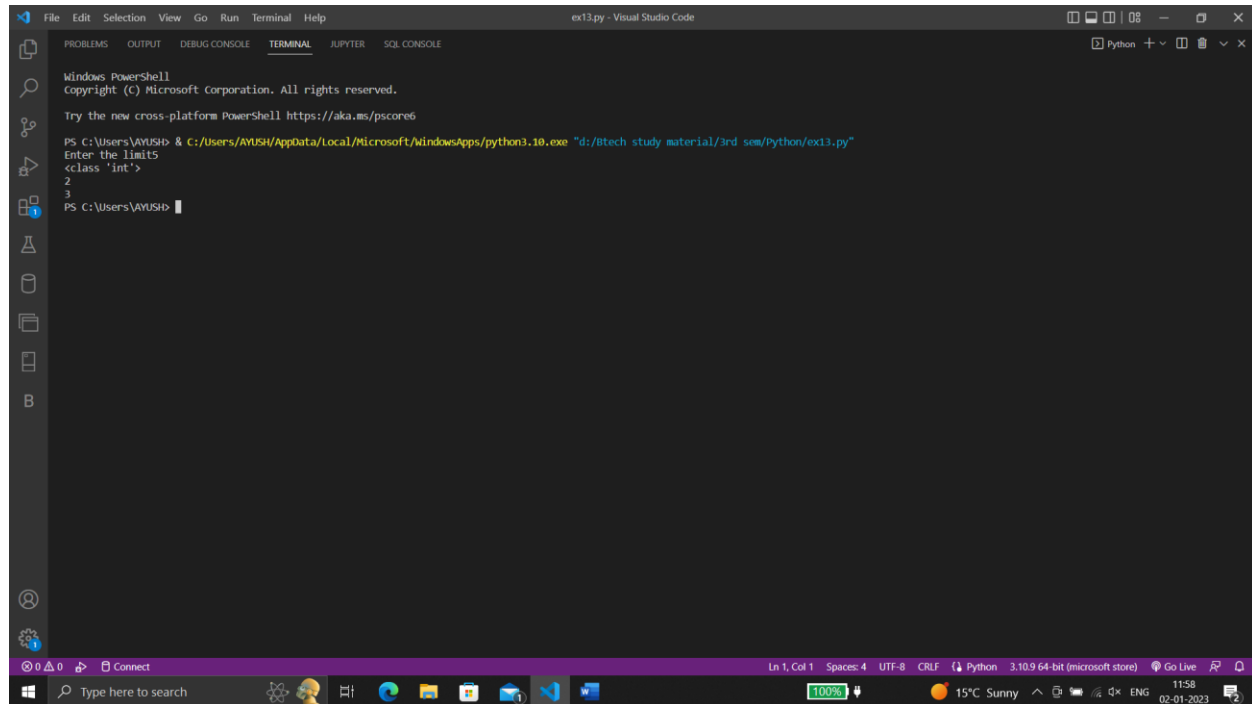
Code:



```
File Edit Selection View Go Run Terminal Help
ex13.py - Visual Studio Code
ex6.py ex7.py ex8.py ex9.py ex10.py ex11.py ex12.py ex13.py x ex14.py
D:\> Btech study material > 3rd sem > Python > ex13.py > n
1 n=int(input("Enter the limit"))
2 print(type(n))
3 for num in range(2, n):
4     if num > 1:
5         for i in range(2, num):
6             if (num % i) == 0:
7                 break
8         else:
9             print(num)
```

The screenshot shows the Visual Studio Code interface with a Python file named ex13.py. The code defines a function to find prime numbers between 2 and a user-defined limit 'n'. It uses nested loops: an outer loop for 'num' from 2 to 'n', and an inner loop for 'i' from 2 to 'num'. If 'num' is divisible by 'i', it breaks the inner loop. If the inner loop completes without breaking, 'num' is a prime and is printed. The status bar at the bottom indicates the file is at Line 1, Column 1, with 4 spaces, using UTF-8 encoding and CRLF line endings. The system tray shows the date as 02-01-2023 and time as 11:57.

Input/Output:



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\AYUSH> & C:/Users/AYUSH/AppData/Local/Microsoft/WindowsApps/python3.10.exe "d:/Btech study material/3rd sem/Python/ex13.py"
Enter the limits
<class 'int'>
2
3
PS C:\Users\AYUSH>
```

LAB EXERCISE 14

Aim:-Implementation of Data Science concepts using Python

DESCRIPTION:-This Data Science with Python program provides learners with a complete understanding of data analytics tools & techniques. Getting started with Python can help you gain knowledge on data analysis, visualization, NumPy, SciPy, web scraping, and natural language processing.

Python offers a good number of libraries used in data science such as Pandas, Numpy, and Scikit-learn. Learning those libraries right away and skipping the basics isn't good though.

If you would like to learn Python for data science, you should master Python core concepts first. Having a solid foundation in Python will help you avoid common mistakes and bad practices. As a result, learning Python libraries used in data science will be much easier.

In this guide, we'll see some must-know Python concepts every data scientist should know. At the end of this article, you will find a Python for Data Science Cheat Sheet in PDF version (section 9 in the table of contents below)

1. Python Attributes vs Methods

I can't tell how long I used the words "attribute" and "method" interchangeably when I was a beginner in Python.

When you learn libraries like Pandas, you'll frequently call attributes and methods, so it's good to know what's the difference between them.

- **Attribute:** An attribute is a variable stored in a class. That is, a value associated with an object. It's referenced by name using dotted expressions. For an object `Foo`, you call an attribute `bar` as `Foo.bar`
- **Method:** A method is a function that is defined inside a class body. For an object `Foo`, you call a method `baz` as `Foo.baz()`