

PRACTICAL FILE
OF
PYTHON PROGRAMMING

BACHELOR OF COMPUTER APPLICATIONS (BCA)

Submitted by:

Name: LAKSH CHIKARA
(Roll No): 24/SCA/BCA(AI-ML)/035



**School of Computer Applications
Manav Rachna International Institute of Research and Studies
(DEEMED TO BE UNIVERSITY)**

Sector-43, Aravalli Hills
Faridabad – 121001

October 2025

Ques: 1) Write a Python program to calculate number of days between two dates. Sample dates: (2014, 7, 2), (2014, 7,11) Expected output : 9 days.

The screenshot shows a code editor interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Search Bar:** sh
- Explorer Bar (Left):** Shows a project structure for a "SHOPPING SITE".
 - __pycache__
 - instance
 - Python
 - practical file.py (selected)
 - static
 - css
 - # style.css
 - images
 - js
 - JS stripe.js
 - templates
 - base.html
 - cart.html
 - checkout.html
 - home.html
 - index.html
 - login.html
 - signup.html
 - thankyou.html
 - app.py
 - models.py
 - requirements.txt
- Code Editor (Center):** A Python script named "practical file.py" with the following code:

```
from datetime import date
date1 = date(2014, 7, 2)
date2 = date(2014, 7, 11)
delta = date2 - date1
print(f"{delta.days} days")
```
- Terminal (Bottom):** Shows command-line output.
 - PS C:\Users\Laksh Chikara\Desktop\shopping site> & "C:/Users/Laksh Chikara/AppData/Local/Temp/ipykernel_1000/1000.py"
9 days
 - PS C:\Users\Laksh Chikara\Desktop\shopping site>
- Bottom Icons:** OUTLINE, TIMELINE, and other standard editor icons.

Ques:2) Write a Python program that accepts an integer (n) and computes the value of n+nn+nnn.

The screenshot shows the Visual Studio Code interface. On the left is the Explorer sidebar, which lists files and folders in the current workspace. The 'practical file.py' file is selected. The main editor area contains the following Python code:

```
1  n = int(input("Enter a number: "))
2  total = n + int(str(n)*2) + int(str(n)*3)
3  print(total)
4  1
```

Below the editor is the Terminal tab, which displays the command-line output of running the script:

```
● PS C:\Users\Laksh Chikara\Desktop\shopping site> & "C:/Users/Laksh Chikara/AppD
Enter a number: 12
122436
○ PS C:\Users\Laksh Chikara\Desktop\shopping site> |
```

The status bar at the bottom shows icons for outline, timeline, and search, along with their respective counts: 1 for outline and 0 for timeline and search.

Ques:3) Ask the user for a number. Depending on whether the number is even or odd, print out an appropriate message to the user. Hint: how does an even / odd number react differently when divided by 2?

The screenshot shows the Visual Studio Code interface. On the left is the Explorer sidebar, which displays a project structure for a 'SHOPPING SITE'. The 'practical file.py' file is selected in the list. The main editor area shows the following Python code:

```
1 num = int(input("Enter a number: "))
2 if num % 2 == 0:
3     print("The number is even.")
4 else:
5     print("The number is odd.")
```

The terminal at the bottom shows the execution of the script and its output:

```
● PS C:\Users\Laksh Chikara\Desktop\shopping site> & "C:/Users/Laksh Chikara/">
Enter a number: 440
The number is even.
○ PS C:\Users\Laksh Chikara\Desktop\shopping site> |
```

Ques:4) Write a Python program which accepts a sequence of comma-separated numbers from user and generate a list and a tuple with those numbers.

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer (Left):** Shows a project structure for "SHOPPING SITE". The "practical file.py" file is selected in the "Python" folder.
- Code Editor (Top Right):** Displays the Python code:

```
1 values = input("Enter comma-separated numbers: ")
2 list_values = values.split(',')
3 tuple_values = tuple(list_values)
4 print(list_values)
5 print(tuple_values)
6
```
- Terminal (Bottom Right):** Shows the output of running the script:

```
● PS C:\Users\Laksh Chikara\Desktop\shopping site> & "C:/Users/Laksh Chikara/">
Enter comma-separated numbers: 10,20,30,40,50
['10', '20', '30', '40', '50']
('10', '20', '30', '40', '50')
○ PS C:\Users\Laksh Chikara\Desktop\shopping site> |
```
- Status Bar (Bottom Left):** Shows icons for outline, timeline, and search, along with the count "0" for both.

Ques:5) Write a Python program to calculate the sum of three given numbers, if the values are equal then return thrice of their sum.

The screenshot shows the Visual Studio Code interface. The left sidebar has a tree view titled 'EXPLORER' under 'SHOPPING SITE'. It lists several files: '_pycache_', 'instance', 'Python', 'practical file.py' (which is selected), 'static', 'css', '# style.css', 'images', 'js' (with 'stripe.js'), 'templates' (containing 'base.html', 'cart.html', 'checkout.html', 'home.html', 'index.html', 'login.html', 'signup.html', 'thankyou.html'), 'app.py', 'models.py', and 'requirements.txt'. The main editor area contains the following Python code:

```
1 def sum_three(x, y, z):
2     if x == y == z:
3         return 3 * (x + y + z)
4     else:
5         return x + y + z
6 print(sum_three(1, 2, 3))
7 print(sum_three(2, 2, 2))
```

Below the editor, there are tabs for 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL', and 'PORTS'. The 'TERMINAL' tab is active, showing the output of running the script:

```
● PS C:\Users\Laksh Chikara\Desktop\shopping site> & "C:/Users/Laksh Chikara/AppData/Local
6
18
○ PS C:\Users\Laksh Chikara\Desktop\shopping site>
```

The bottom left corner shows status icons for 'OUTLINE' and 'TIMELINE', and a message '0 ▲ 0'.

Ques:6) Write a Python program to test whether a passed letter is a vowel or not

The screenshot shows a code editor interface with a sidebar and a main workspace.

Sidebar:

- SHOPPING SITE
 - _pycache_
 - instance
 - Python
 - practical file.py
 - static
 - css
 - # style.css
 - images
 - js
 - stripe.js
 - templates
 - base.html
 - cart.html
 - checkout.html
 - home.html
 - index.html
 - login.html
 - signup.html
 - thankyou.html
 - app.py
 - models.py
 - requirements.txt

Main Workspace:

```
Python > practical file.py > ...
1  letter = input("Enter a letter: ").lower()
2  if letter in 'aeiou':
3      print("Vowel")
4  else:
5      print("Not a vowel")
6
```

Bottom Navigation:

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

● PS C:\Users\Laksh Chikara\Desktop\shopping site> & "C:/Users/Laksh Chikara/AppData/Local/Programs/Python/Python39/python.exe" "C:/Users/Laksh Chikara/Desktop/shopping site/practical file.py
Enter a letter: I
Vowel
○ PS C:\Users\Laksh Chikara\Desktop\shopping site> |

> OUTLINE
> TIMELINE
0 △ 0

Ques:7) Take a list, say for example this one:

a = [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89]

and write a program that prints out all the elements of the list that are less than 5.

Extras:

- a) Instead of printing the elements one by one, make a new list that has all the elements less than 5 from this list in it and print out this new list.
- b) Write this in one line of Python.
- c) Ask the user for a number and return a list that contains only elements from the original list a that are smaller than that number given by the user.

The screenshot shows a code editor interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Explorer:** Shows a project structure for a "SHOPPING SITE".
 - Python folder:
 - practical file.py
 - static folder:
 - css folder:
 - # style.css
 - images folder
 - js folder:
 - stripe.js
 - templates folder:
 - base.html
 - cart.html
 - checkout.html
 - home.html
 - index.html
 - login.html
 - signup.html
 - thankyou.html
 - app.py
 - models.py
 - requirements.txt
- Code Editor:** The file "practical file.py" is open, containing the following Python code:

```
1 a = [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89]
2 # a) Print elements less than 5
3 for x in a:
4     if x < 5:
5         print(x)
6
7 # b) New List (one line)
8 print([x for x in a if x < 5])
9
10 # c) Filter by user-given number
11 num = int(input("Enter a number: "))
12 print([x for x in a if x < num])
```
- Terminal:** Shows the output of running the script.

```
PS C:\Users\Laksh Chikara\Desktop\shopping site> & "C:/Users/Laksh Chikara/AppData/Local/Programs/Python/Python39/python.exe" practical file.py
1
1
2
3
[1, 1, 2, 3]
Enter a number:
```
- Bottom Bar:** PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL (underlined), PORTS.
- Status Bar:** Shows icons for profile, outline, timeline, and file status (0 changes).

Ques:8) Create a program that asks the user for a number and then prints out a list of all the divisors of that number. (If you don't know what a divisor is, it is a number that divides evenly into another number. For example, 13 is a divisor of 26 because 26 / 13 has no remainder.)

The screenshot shows the Visual Studio Code interface. The left sidebar displays a file tree for a project named "SHOPPING SITE". The "practical file.py" file is selected. The main editor area contains the following Python code:

```
1 num = int(input("Enter a number: "))
2 divisors = []
3 for i in range(1, num + 1):
4     if num % i == 0:
5         divisors.append(i)
6 print(divisors)
7
```

The bottom right panel shows the terminal output:

```
● PS C:\Users\Laksh Chikara\Desktop\shopping site> & "C:/Users/Laksh Chikara/Applications/PyCharm.app/Contents/bin/python" "C:/Users/Laksh Chikara/Desktop/shopping site/practical file.py"
Enter a number: 07
[1, 7]
```

The status bar at the bottom indicates there are 1 error and 0 warnings.

Ques:9) Take two lists, say for example these two:

a = [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89]

b = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13]

and write a program that returns a list that contains only the elements that are common between the lists (without duplicates). Make sure your program works on two lists of different sizes.

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows a project structure for a "SHOPPING SITE". It includes Python files like "practical file.py", "app.py", and "models.py", as well as static files like "style.css" and "stripe.js", and HTML files like "base.html", "cart.html", etc.
- Code Editor:** Displays the content of "practical file.py". The code uses set intersection to find common elements between lists a and b.
- Terminal:** Shows the output of running the script, which prints the common elements [1, 2, 3, 5, 8, 13].
- Status Bar:** Shows icons for outline, timeline, and other development tools.

```
1 a = [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89]
2 b = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13]
3 common = list(set(a) & set(b))
4 print(common)
5
```

```
● PS C:\Users\Laksh Chikara\Desktop\shopping site> & "C:/Users/Laksh Chikara/AppData/Local[1, 2, 3, 5, 8, 13]
○ PS C:\Users\Laksh Chikara\Desktop\shopping site>
```

Ques:10) Ask the user for a string and print out whether this string is a palindrome or not. (A palindrome is a string that reads the same forwards and backwards.)

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer (Left):** Shows a project structure for a "SHOPPING SITE". The "practical file.py" file is selected. Other files include __pycache__, instance, Python, static, css, js (with stripe.js), templates, and various HTML files like base.html, cart.html, checkout.html, home.html, index.html, login.html, signup.html, and thankyou.html. There are also app.py, models.py, and requirements.txt.
- Code Editor (Center):** Displays the content of practical file.py:

```
1 s = input("Enter a string: ")
2 if s == s[::-1]:
3     print("Palindrome")
4 else:
5     print("Not a palindrome")
```
- Terminal (Bottom):** Shows the output of running the script:

```
PS C:\Users\Laksh Chikara\Desktop\shopping site> & "C:/Users/Laksh Chikara/AppData/Local/Programs/Python/Python38-32/practical file.py"
Enter a string: JAAT
Not a palindrome
PS C:\Users\Laksh Chikara\Desktop\shopping site> & "C:/Users/Laksh Chikara/AppData/Local/Programs/Python/Python38-32/practical file.py"
Enter a string: CNC
Palindrome
PS C:\Users\Laksh Chikara\Desktop\shopping site> |
```

Ques:11) Let's say I give you a list saved in a variable: a = [1, 4, 9, 16, 25, 36, 49, 64, 81, 100]. Write one line of Python that takes this list a and makes a new list that has only the even elements of this list in it.

The screenshot shows the Visual Studio Code interface. The left sidebar displays a file tree for a project named "SHOPPING SITE". The "practical file.py" file is selected in the Explorer view. The code editor window contains the following Python code:

```
1 a = [1, 4, 9, 16, 25, 36, 49, 64, 81, 100]
2 even = [x for x in a if x % 2 == 0]
3 print(even)
4 
```

The terminal window at the bottom shows the output of running the script:

```
● PS C:\Users\Laksh Chikara\Desktop\shopping site> & "C:/Users/Laksh Chikara/AppData
[4, 16, 36, 64, 100]
○ PS C:\Users\Laksh Chikara\Desktop\shopping site>
```