PYTHON PROGRAMS ASSIGNMENT 1

ARTHIMETIC OPERATORS

main.py

```python
a = 46     # Initializing the value of a
b = 4      # Initializing the value of b

print("For a =", a, "and b =", b,"\nCalculate the following:")

# printing different results
print('1. Addition of two numbers: a + b =', a + b)
print('2. Subtraction of two numbers: a - b =', a - b)
print('3. Multiplication of two numbers: a * b =', a * b)
print('4. Division of two numbers: a / b =', a / b)
print('5. Floor division of two numbers: a // b =',a // b)
print('6. Reminder of two numbers: a mod b =', a % b)
print('7. Exponent of two numbers: a ^ b =',a ** b)
```

Output

```
For a = 46 and b = 4
Calculate the following:
1. Addition of two numbers: a + b = 50
2. Subtraction of two numbers: a - b = 42
3. Multiplication of two numbers: a * b = 184
4. Division of two numbers: a / b = 11.5
5. Floor division of two numbers: a // b = 11
6. Reminder of two numbers: a mod b = 2
7. Exponent of two numbers: a ^ b = 4477456

=== Code Execution Successful ===
```

COMPARISON OPERATORS

main.py

```python
a = 46     # Initializing the value of a
b = 4      # Initializing the value of b

print("For a =", a, "and b =", b,"\nCheck the following:")

# printing different results
print('1. Two numbers are equal or not:', a == b)
print('2. Two numbers are not equal or not:', a != b)
print('3. a is less than or equal to b:', a <= b)
print('4. a is greater than or equal to b:', a >= b)
print('5. a is greater b:', a > b)
print('6. a is less than b:', a < b)
```

Output

```
For a = 46 and b = 4
Check the following:
1. Two numbers are equal or not: False
2. Two numbers are not equal or not: True
3. a is less than or equal to b: False
4. a is greater than or equal to b: True
5. a is greater b: True
6. a is less than b: False

=== Code Execution Successful ===
```

ASSIGNMENT OPEARATORS

main.py

```python
a = 34          # Initialize the value of a
b = 6           # Initialize the value of b

# printing the different results
print('a += b:', a + b)
print('a -= b:', a - b)
print('a *= b:', a * b)
print('a /= b:', a / b)
print('a %= b:', a % b)
print('a **= b:', a ** b)
print('a //= b:', a // b)
```

Output

```
a += b: 40
a -= b: 28
a *= b: 204
a /= b: 5.666666666666667
a %= b: 4
a **= b: 1544804416
a //= b: 5

=== Code Execution Successful ===
```

BITWISE OPERATORS

main.py

```python
a = 7           # initializing the value of a
b = 8           # initializing the value of b

# printing different results
print('a & b :', a & b)
print('a | b :', a | b)
print('a ^ b :', a ^ b)
print('~a :', ~a)
print('a << b :', a << b)
print('a >> b :', a >> b)
```

Output

```
a & b : 0
a | b : 15
a ^ b : 15
~a : -8
a << b : 1792
a >> b : 0

=== Code Execution Successful ===
```

Logical Operators

```python
a = 7          # initializing the value of a

# printing different results
print("For a = 7, checking whether the following conditions are True or False:")
print('\"a > 5 and a < 7\" =>', a > 5 and a < 7)
print('\"a > 5 or a < 7\" =>', a > 5 or a < 7)
print('\"not (a > 5 and a < 7)\" =>', not(a > 5 and a < 7))
```

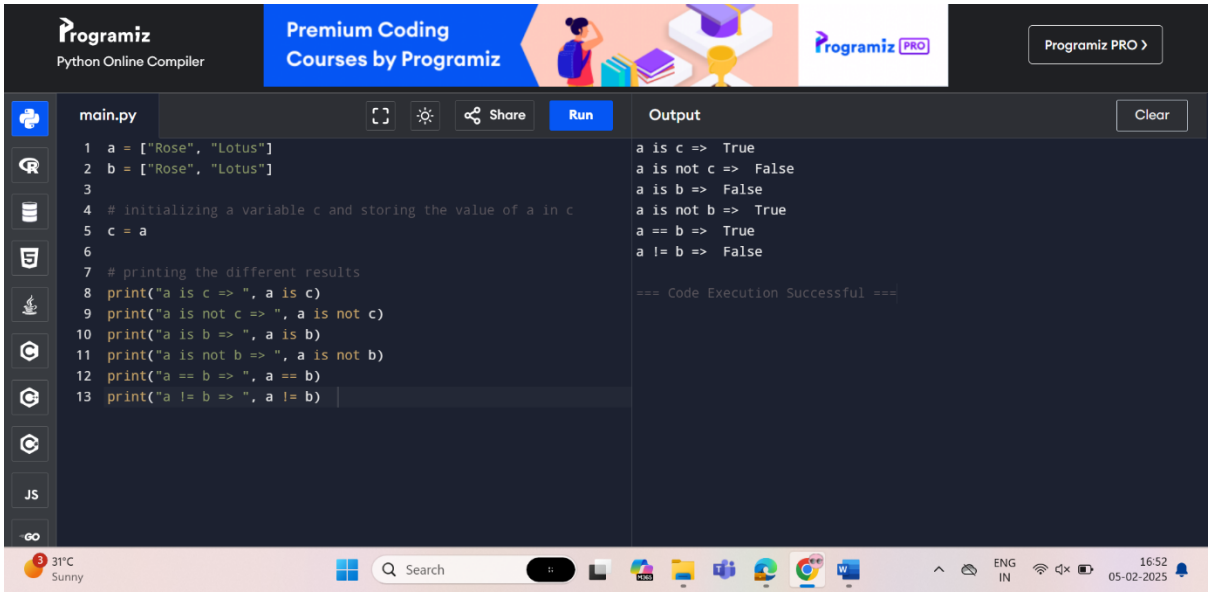
```
For a = 7, checking whether the following conditions are True or False:
"a > 5 and a < 7" => False
"a > 5 or a < 7" => True
"not (a > 5 and a < 7)" => True

=== Code Execution Successful ===
```

MEMBERSHIP OPERATORS

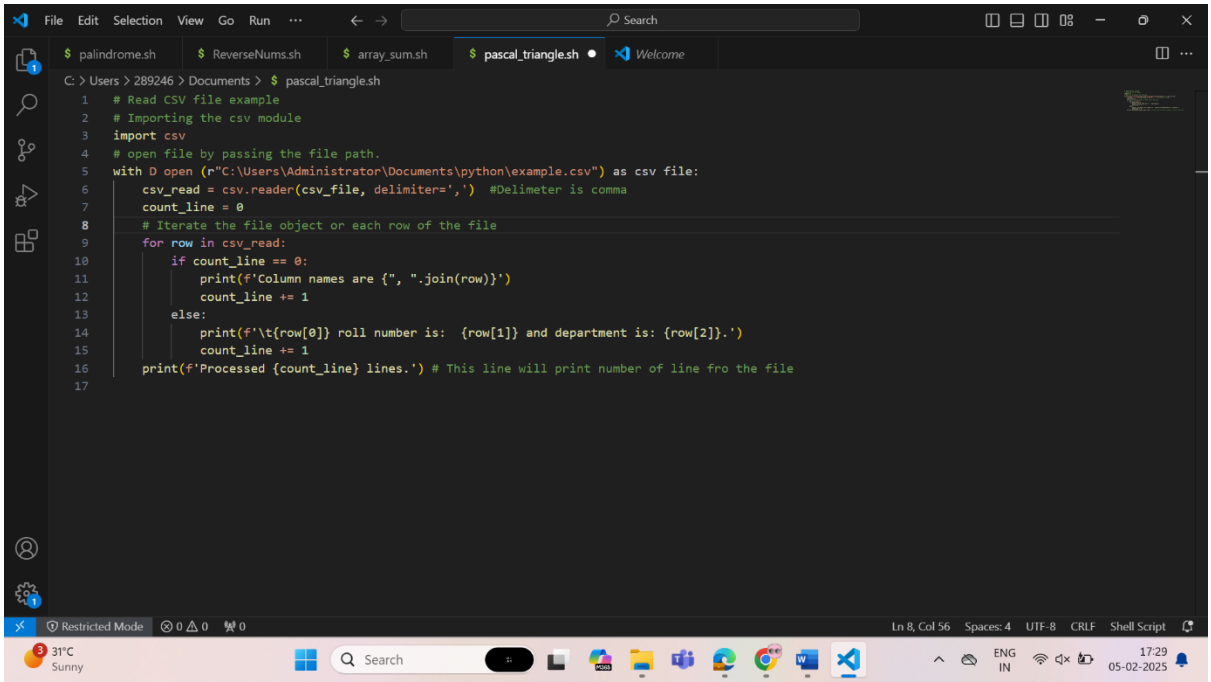
```python
# initializing a list
myList = [12, 22, 28, 35, 42, 49, 54, 65, 92, 103, 245, 874]

x = 31
y = 28

print("Given List:", myList)

if (x not in myList):
    print("x =", x,"is NOT present in the given list.")
else:
    print("x =", x,"is present in the given list.")

if (y in myList):
    print("y =", y,"is present in the given list.")
else:
    print("y =", y,"is NOT present in the given list.")
```

```
a += b: 40
a -= b: 28
a *= b: 204
a /= b: 5.666666666666667
a %= b: 4
a **= b: 1544804416
a //= b: 5

=== Code Execution Successful ===
```

IDENTITY OPERATOR



How to read CSV file in Python



OUTPUT:-

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

 Files\Python313\python.exe' 'c:\Users\Administrator\.vscode\extensions\ms-python.debugpy-2024.14.0-wi
n32-x64\bundled\libs\debugpy\adapter/../..\debugpy\launcher' '59728' '--' 'c:\Users\Administrator\reci
pewebsite\import_csv_module.py'
Column names are Name, Roll Number, Department
        Alice roll number is:  101 and department is: Computer Science.
        Bob roll number is:  102 and department is: Mechanical.
        Charlie roll number is:  103 and department is: Electrical.
        David roll number is:  104 and department is: Civil.
        Emma roll number is:  105 and department is: Electronics.
Processed 6 lines.
PS C:\Users\Administrator\recipewebsite>
```

REVERSE A STRING PYTHON

USING FOR LOOP

```
main.py                                   Share   Run        Output                    Clear
1 def reverse_string(str):                          The original string is:  JavaTpoint
2     str1 = ""                                     The reverse string is tniopTavaJ
3     for i in str:
4         str1 = i + str1                           === Code Execution Successful ===
5     return str1
6
7 str = "JavaTpoint"
8 print("The original string is: ",str)
9 print("The reverse string is",reverse_string(str))
```

USING WHILE LOOP

main.py

```python
str = "JavaTpoint" #  string variable
print ("The original string  is : ",str)
reverse_String = ""  # Empty String
count = len(str) # Find length of a string and save in count variable
while count > 0:
    reverse_String += str[ count - 1 ] # save the value of str[count-1] in reverseString
    count = count - 1 # decrement index
print ("The reversed string using a while loop is : "
    ,reverse_String)# reversed string
```

Output

```
The original string  is :  JavaTpoint
The reversed string using a while loop is :  tniopTavaJ

=== Code Execution Successful ===
```

USING SLICE OPERATOR

main.py

```python
def reverse(str):
    str = str[::-1]
    return str

s = "JavaTpoint"
print ("The original string  is : ",s)
print ("The reversed string using extended slice operator  is : "
    ,reverse(s))
```

Output

```
The original string  is :  JavaTpoint
The reversed string using extended slice operator  is :  tniopTavaJ

=== Code Execution Successful ===
```

Using reverse function with join

main.py

```python
def reverse(str):
    string = "".join(reversed(str)) # reversed() function inside the join() function
    return string

s = "JavaTpoint"

print ("The original string is : ",s)
print ("The reversed string using reversed() is : ",reverse(s) )
```

Output

```
The original string is :  JavaTpoint
The reversed string using reversed() is :  tniopTavaJ

=== Code Execution Successful ===
```

Using recursion()

```python
def reverse(str):
    if len(str) == 0: # Checking the lenght of string
        return str
    else:
        return reverse(str[1:]) + str[0]

str = "Devansh Sharma"
print ("The original string  is : ", str)
print ("The reversed string(using recursion) is : ", reverse(str))
```

```
The original string  is :  Devansh Sharma
The reversed string(using recursion) is :  amrahS hsnaveD

=== Code Execution Successful ===
```

PYTHON IF ELSE STATEMENTS

Simple Python program to understand the if statement

```python
num = int(input("enter the number:"))
# Here, we are taking an integer num and taking input dynamically

if num%2 == 0:
# Here, we are checking the condition. If the condition is true, we will enter the block
    print("The Given number is an even number")
```

```
enter the number:4
The Given number is an even number

=== Code Execution Successful ===
```

Program to print the largest of the three numbers.

```python
# Simple Python Program to print the largest of the three numbers
a = int (input("Enter a: "));
b = int (input("Enter b: "));
c = int (input("Enter c: "));
if a>b and a>c:
    print ("From the above three numbers given a is largest");

if b>a and b>c:
    print ("From the above three numbers given b is largest");

if c>a and c>b:
    print ("From the above three numbers given c is largest");
```

```
Enter a: 5
Enter b: 6
Enter c: 3
From the above three numbers given b is largest

=== Code Execution Successful ===
```

Program to check whether a person is eligible to vote or not.

```python
age = int (input("Enter your age: "))
# Here, we are taking an integer num and taking input dynamically

if age>=18:
# Here, we are checking the condition. If the condition is true, we will enter the block
    print("You are eligible to vote !!");
else:
    print("Sorry! you have to wait !!");
```

```
Enter your age: 21
You are eligible to vote !!

=== Code Execution Successful ===
```

Program to check whether a number is even or not

```python
number = int(input("Enter the number?"))
if number==10:

    print("The given number is equals to 10")
elif number==50:

    print("The given number is equal to 50");
elif number==100:
    print("The given number is equal to 100");
else:
    print("The given number is not equal to 10, 50 or 100");
```

```
Enter the number?50
The given number is equal to 50

=== Code Execution Successful ===
```

Simple Python program to understand elif statement

```python
number = int(input("Enter the number?"))
if number==10:

    print("The given number is equals to 10")
elif number==50:

    print("The given number is equal to 50");
elif number==100:
    print("The given number is equal to 100");
else:
    print("The given number is not equal to 10, 50 or 100");
```

```
Enter the number?50
The given number is equal to 50

=== Code Execution Successful ===
```

PYTHON FOR LOOP

Python program to show how the for loop works

main.py

```python
# Python program to show how the for loop works

# Creating a sequence which is a tuple of numbers
numbers = [4, 2, 6, 7, 3, 5, 8, 10, 6, 1, 9, 2]

# variable to store the square of the number
square = 0

# Creating an empty list
squares = []

# Creating a for loop
for value in numbers:
    square = value ** 2
    squares.append(square)
print("The list of squares is", squares)
```

Output

```
The list of squares is [16, 4, 36, 49, 9, 25, 64, 100, 36, 1, 81, 4]

=== Code Execution Successful ===
```

Python program to show how if-else statements work

main.py

```python

string = "Python Loop"

# Initiating a loop
for s in string:
    # giving a condition in if block
    if s == "o":
        print("If block")
    # if condition is not satisfied then else block will be executed
    else:
        print(s)
```

Output

```
P
y
t
h
If block
n

L
If block
If block
p

=== Code Execution Successful ===
```

Python program to show the working of range() function

main.py

```python
# Python program to show the working of range() function

print(range(15))

print(list(range(15)))

print(list(range(4, 9)))

print(list(range(5, 25, 4)))
```

Output

```
range(0, 15)
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14]
[4, 5, 6, 7, 8]
[5, 9, 13, 17, 21]

=== Code Execution Successful ===
```

Python program to iterate over a sequence with the help of indexing

```python
# Python program to iterate over a sequence with the help of indexing

tuple_ = ("Python", "Loops", "Sequence", "Condition", "Range")

# iterating over tuple_ using range() function
for iterator in range(len(tuple_)):
    print(tuple_[iterator].upper())
```

```
PYTHON
LOOPS
SEQUENCE
CONDITION
RANGE

=== Code Execution Successful ===
```

PYTHON WHILE LOOP

Python program to show how to use a while loop

```python
# Python program to show how to use a while loop
counter = 0
# Initiating the loop
while counter < 10: # giving the condition
    counter = counter + 3
    print("Python Loops")
```

```
Python Loops
Python Loops
Python Loops
Python Loops

=== Code Execution Successful ===
```

Using else Statement with while Loops

```python
#Python program to show how to use else statement with the while loop
counter = 0

# Iterating through the while loop
while (counter < 10):
    print("Python Loops") # Executed untile condition is met
    counter = counter + 3
# Once the condition of while loop gives False this statement will be executed
else:
    print("Code block inside the else statement")
```

```
Python Loops
Python Loops
Python Loops
Python Loops
Code block inside the else statement

=== Code Execution Successful ===
```

Continue Statement

Python program to show how the continue statement works

```python
# Python program to show how the continue statement works

# Initiating the loop
for string in "Python Loops":
    if string == "o" or string == "p" or string == "t":
        continue
    print('Current Letter:', string)
```

```
Current Letter: P
Current Letter: y
Current Letter: h
Current Letter: n
Current Letter:
Current Letter: L
Current Letter: s

=== Code Execution Successful ===
```

Break Statement

It stops the execution of the loop when the break statement is reached.

Code:-

```python
# Python program to show how the break statement works

# Initiating the loop
for string in "Python Loops":
    if string == 'L':
        break
    print('Current Letter: ', string)
```

```
Current Letter:  P
Current Letter:  y
Current Letter:  t
Current Letter:  h
Current Letter:  o
Current Letter:  n
Current Letter:

=== Code Execution Successful ===
```

PYTHON STRINGS

Creating String in Python

```python
#Using single quotes
str1 = 'Hello Python'
print(str1)
#Using double quotes
str2 = "Hello Python"
print(str2)

#Using triple quotes
str3 = '''''Triple quotes are generally used for
    represent the multiline or
    docstring'''
print(str3)
```

```
Hello Python
Hello Python
''Triple quotes are generally used for
    represent the multiline or
    docstring

=== Code Execution Successful ===
```

Strings indexing and splitting

Example 1:

main.py

```python
str = "HELLO"
print(str[0])
print(str[1])
print(str[2])
print(str[3])
print(str[4])
# It returns the IndexError because 6th index doesn't exist
print(str[6])
```

Output

```
H
E
L
L
O
ERROR!
Traceback (most recent call last):
  File "<main.py>", line 8, in <module>
IndexError: string index out of range

=== Code Exited With Errors ===
```

Example 2

main.py

```python

str = "JAVATPOINT"
print(str[0:])
# Starts 1th index to 4th index
print(str[1:5])
# Starts 2nd index to 3rd index
print(str[2:4])
# Starts 0th to 2nd index
print(str[:3])
#Starts 4th to 6th index
print(str[4:7])
```

Output

```
JAVATPOINT
AVAT
VA
JAV
TPO

=== Code Execution Successful ===
```

Strings Operators

main.py

```python
str = "Hello"
str1 = " world"
print(str*3) # prints HelloHelloHello
print(str+str1)# prints Hello world
print(str[4]) # prints o
print(str[2:4]); # prints ll
print('w' in str) # prints false as w is not present in str
print('wo' not in str1) # prints false as wo is present in str1.

print(r'C://python37') # prints C://python37 as it is written
print("The string str : %s"%(str)) # prints The string str :
    Hello
```

Output

```
HelloHelloHello
Hello world
o
ll
False
False
C://python37
The string str : Hello

=== Code Execution Successful ===
```

String formatting

```python
# Using Curly braces
print("{} and {} both are the best friend".format("Devansh"
    ,"Abhishek"))

#Positional Argument
print("{1} and {0} best players ".format("Virat","Rohit"))

#Keyword Argument
print("{a},{b},{c}".format(a = "James", b = "Peter", c = "Ricky"))
```

```
Devansh and Abhishek both are the best friend
Rohit and Virat best players
James,Peter,Ricky

=== Code Execution Successful ===
```

List and Tuple Syntax Differences

```python
# Python code to show the difference between creating a list and
    a tuple

list_ = [4, 5, 7, 1, 7]
tuple_ = (4, 1, 8, 3, 9)

print("List is: ", list_)
print("Tuple is: ", tuple_)
```

```
List is:  [4, 5, 7, 1, 7]
Tuple is:  (4, 1, 8, 3, 9)

=== Code Execution Successful ===
```

Updating the element of list and tuple at a particular index

```python

# creating a list and a tuple
list_ = ["Python", "Lists", "Tuples", "Differences"]
tuple_ = ("Python", "Lists", "Tuples", "Differences")

list_[3] = "Mutable"
print( list_ )
try:
    tuple_[3] = "Immutable"
    print( tuple_ )
except TypeError:
    print( "Tuples cannot be modified because they are immutable"
        )
```

```
['Python', 'Lists', 'Tuples', 'Mutable']
Tuples cannot be modified because they are immutable

=== Code Execution Successful ===
```

Code to show the difference in the size of a list and a tuple