

PYTHON ASSIGNMENTS D16

The task for class conducted on 25-08-2023

Task 1: Print function

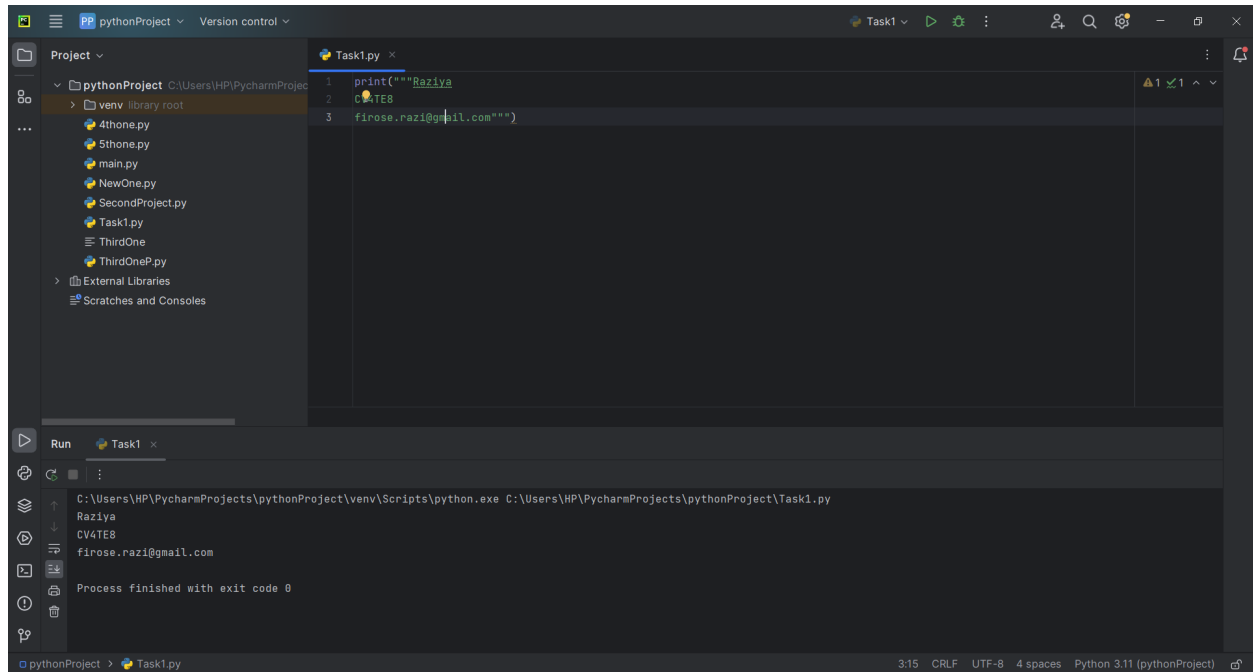
Python code that prints your name, student number and email address.

An example runs of the program:

Bob

ST1001

bob@gmail.com



The screenshot shows the PyCharm IDE interface. The left sidebar displays the project structure for 'pythonProject', including files like '4thone.py', '5thone.py', 'main.py', 'NewOne.py', 'SecondProject.py', 'Task1.py', 'ThirdOne', and 'ThirdOneP.py'. The main editor window shows the code in 'Task1.py':

```
1 print("Raziya")
2 CV4TE8
3 firose.razi@gmail.com")
```

The bottom panel shows the 'Run' output for 'Task1'. The command executed is 'C:\Users\HP\PycharmProjects\pythonProject\venv\Scripts\python.exe C:\Users\HP\PycharmProjects\pythonProject\Task1.py'. The output is:

```
Raziya
CV4TE8
firose.razi@gmail.com
```

The process finished with exit code 0. The status bar at the bottom indicates the file encoding is UTF-8, 4 spaces, and the Python version is 3.11.

Task 2 The task for class conducted on 01-09-2023

Escape sequences:

Exercise 1:

Write a Python code that displays the numbers from 1 to 5 as steps.

An example runs of the program:

```
1
2
3
4
5
```

Exercise 2:

Write a Python code that outputs the following sentence (including the quotation marks and line break) to the screen:

An example run of the program:

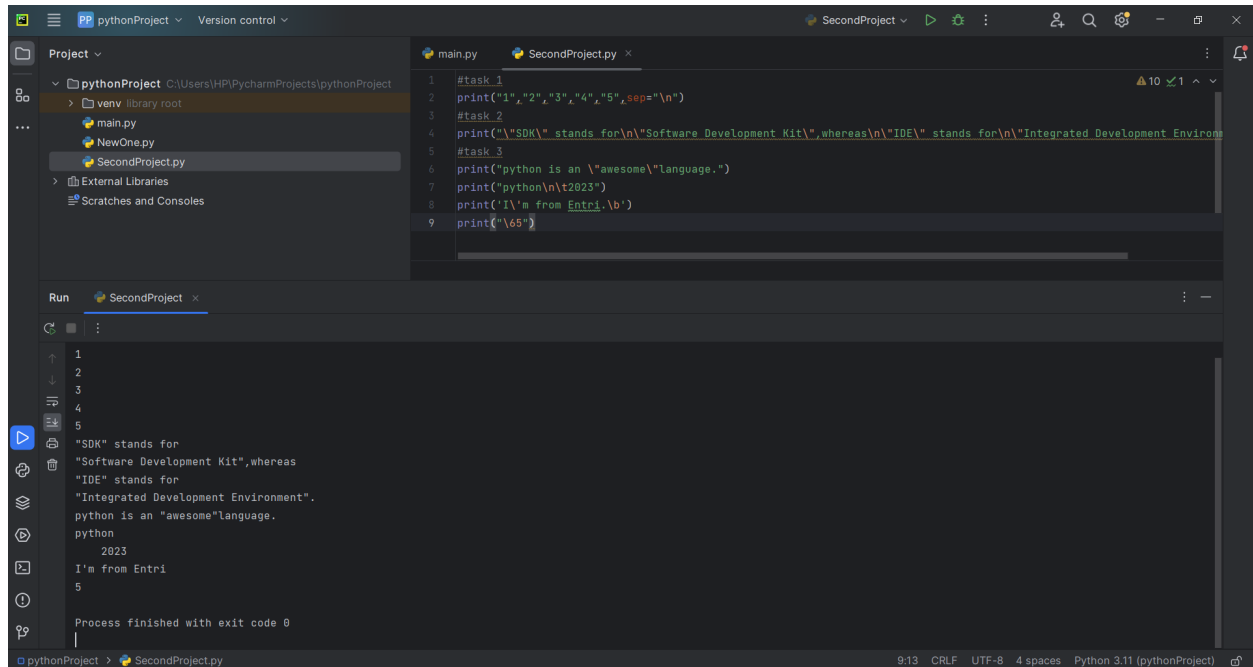
"SDK" stands for

"Software Development Kit", whereas
"IDE" stands for
"Integrated Development Environment".

Exercise 3:

Practice and check the output

```
print("python is an \"awesome\" language.")  
print("python\n\t2023")  
print('I\'m from Entri.\b')  
print("\65")
```



The screenshot shows the PyCharm IDE interface. The top pane displays a Python script named `SecondProject.py` with the following code:

```
1 #task 1  
2 print("1"_"2"_"3"_"4"_"5",sep="\n")  
3 #task 2  
4 print("\nSDK" stands for\n"Software Development Kit",whereas\n"IDE" stands for\n"Integrated Development Environme  
5 #task 3  
6 print("python is an \"awesome\" language.")  
7 print("python\n\t2023")  
8 print('I\'m from Entri.\b')  
9 print("\65")
```

The bottom pane shows the output of the script:

```
1  
2  
3  
4  
5  
"SDK" stands for  
"Software Development Kit",whereas  
"IDE" stands for  
"Integrated Development Environment".  
python is an "awesome"language.  
python  
2023  
I'm from Entri  
5  
Process finished with exit code 0
```

Task 3 The task for class conducted on 04-09-2023

Datatypes, User-interactive command

Exercise 1

Write Python code that asks the user to enter his/her name and then output/prints his/her name with a greeting.

An example runs of the program:

Please enter your name: Tony

Hi Tony, welcome to Python programming :)

Exercise 2

Name your file: **PoundsToDollars.py**

Write a program that asks the user to enter an amount in pounds (£) and the program calculates and converts an amount in dollars (\$)

An example runs of the program:

Please enter the amount in pounds: XXX

£ XXX are \$ XXX

Exercise 3

Name your file: ***MyInfoInput.py***

Write a program that asks the user to enter his/her name, student number, and email address and display it as in the example run.

An example runs of the program:

Enter your name: Bob

Enter your student number: ST1001

Enter your email address: bob@gmail.com

Hello Bob, thanks for entering your details. We will email you at bob@gmail.com with subject id: ST1001 for further information.

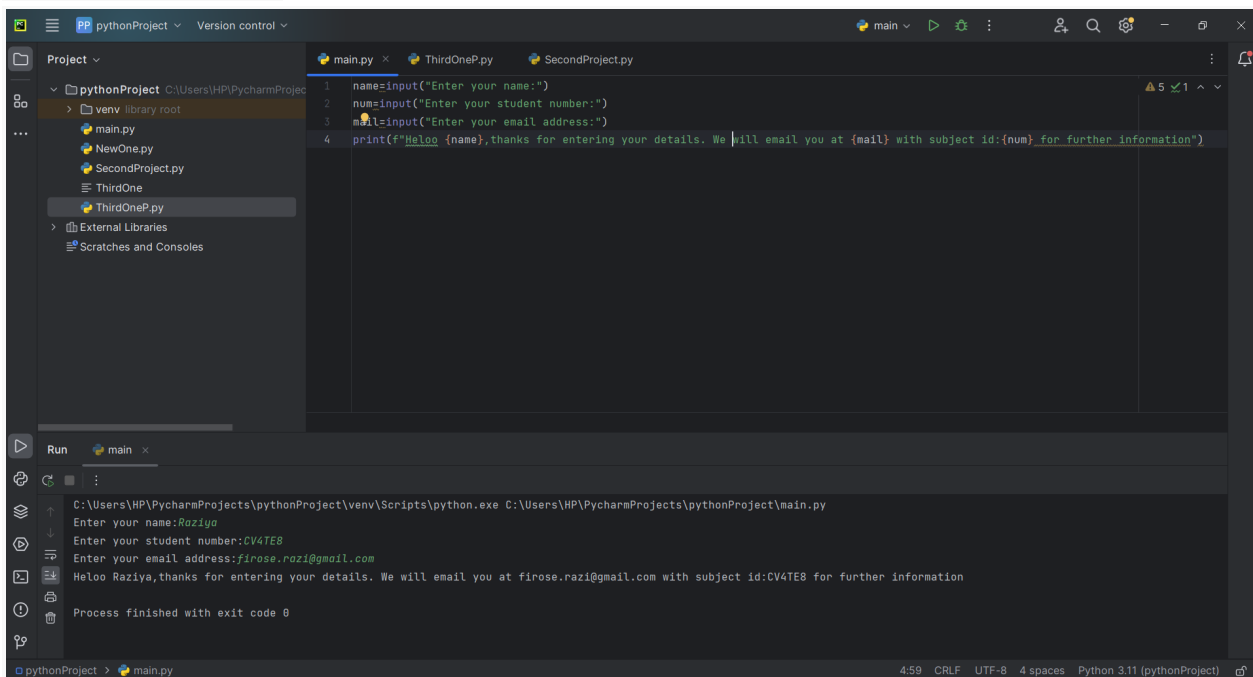
Exercise 4

Define the variables below. Print the types of each variable. What is the sum of your variables? (*Hint: use a type conversion function.*) What data type is the sum?

num=23

textnum="57"

decimal=98.3 (edited)



The screenshot shows the PyCharm IDE interface. The top pane displays the code for `main.py`:

```
1 name=input("Enter your name:")
2 num=input("Enter your student number:")
3 email=input("Enter your email address:")
4 print(f"Hello {name}, thanks for entering your details. We will email you at {email} with subject id:{num} for further information")
```

The bottom pane shows the output of the program:

```
C:\Users\HP\PycharmProjects\pythonProject\venv\Scripts\python.exe C:\Users\HP\PycharmProjects\pythonProject\main.py
Enter your name:Raziya
Enter your student number:CV4TE8
Enter your email address:firose.razi@gmail.com
Hello Raziya, thanks for entering your details. We will email you at firose.razi@gmail.com with subject id:CV4TE8 for further information
Process finished with exit code 0
```

The screenshot shows the PyCharm IDE interface. The top toolbar includes icons for running and debugging. The left sidebar shows the project structure with 'Task3.py' selected. The main editor displays the code for 'Task3.py', which includes two exercises. Exercise 1 prompts the user for their name and prints a welcome message. Exercise 4 defines variables for an integer, a string, and a float, then prints their types and a sum. The bottom pane shows the execution output, which matches the code's behavior: it prompts for a name ('raziya'), prints a welcome message, and then prints the types and sum of the variables defined in Exercise 4.

```
1 # exercise 1
2 name = (input("Please enter your name :"))
3 print("Hi", name, "Welcome to python programming:")
4
5 #exercise 4
6 num = 23 # this is an integer
7 textnum = '57' # this is a string
8 decimal = 98.3 # this is a float
9 print(type(num))
10 print(type(textnum))
11 print(type(decimal))
12 sum = num + int(textnum) + decimal
13 print(sum)
14 print(type(sum))
```

Run Task3

C:\Users\HP\PycharmProjects\pythonProject\venv\Scripts\python.exe C:\Users\HP\PycharmProjects\pythonProject\venv\Task3.py

Please enter your name :raziya

Hi raziya Welcome to python programming:)

<class 'int'>

<class 'str'>

<class 'float'>

178.3

<class 'float'>

Process finished with exit code 0

The screenshot shows the PyCharm IDE interface. The top toolbar includes icons for running and debugging. The left sidebar shows the project structure with 'Task3.py' selected. The main editor displays the code for 'Task3.py', which includes two exercises. Exercise 2 prompts the user for the amount in pounds and prints the equivalent in dollars. Exercise 3 prompts the user for their name and prints a welcome message. The bottom pane shows the execution output, which matches the code's behavior: it prompts for a name ('raziya'), prints a welcome message, then prompts for the amount in pounds (1.0) and prints the equivalent in dollars (\$1.26).

```
6 num = 23 # this is an integer
7 textnum = '57' # this is a string
8 decimal = 98.3 # this is a float
9 print(type(num))
10 print(type(textnum))
11 print(type(decimal))
12 sum = num + int(textnum) + decimal
13 print(sum)
14 print(type(sum))
15
16 #exercise 2
17 pounds = float(input("Please enter the amount in pounds :"))
18 dollars = 1.26 * pounds
19 print("pounds",pounds,"are $",dollars)
```

Run Task3

C:\Users\HP\PycharmProjects\pythonProject\venv\Scripts\python.exe C:\Users\HP\PycharmProjects\pythonProject\venv\Task3.py

Please enter your name :raziya

Hi raziya Welcome to python programming:)

<class 'int'>

<class 'str'>

<class 'float'>

178.3

<class 'float'>

Please enter the amount in pounds :1

pounds 1.0 are \$ 1.26

Process finished with exit code 0

Task 4 Task for class conducted on 08-09-2023

Selection Statements and Loops:

SELECTION STATEMENTS:

Exercise 1

Name your file: MonthNames.py

Write a program that reads an integer value between 1 and 12 from the user and prints output for the corresponding month of the year.

An example run of the program (numbers in bold are typed in by the user)

Enter the month: 3

Month 3 is March

Exercise 2

Name your file: EvenOddNumbers.py

Write a program that reads an integer value from the user and prints output whether it is odd or even.

An example run of the program (numbers in bold are typed in by the user)

Enter a number: 13

13 is "even"

Exercise 3

Name your file: CinemaTickets.py

A certain cinema currently sells tickets for the full price of 6 pounds, but always sells tickets for half price to people who are less than 16 years old, and for a third of the price for people who are 60 years old or more.

An example run of the program (numbers in bold are typed in by the user)

Enter your age: 63

Your ticket costs £2.00

Exercise 4

Name your file: BodyMassIndex.py

Write a program to calculate your BMI and give weight status. Body Mass Index (BMI) is an internationally used measurement to check if you are a healthy weight for your height. The metric BMI formula accepts weight in kilograms and height in meters:

$BMI = \text{weight(kg)} / \text{height}^2(\text{m}^2)$

BMI Weight Status Categories table

BMI range - kg/m ²	Category
Below 18.5	Underweight
18.5 -24.9	Normal
25 - 29.9	Overweight
30 & Above	Obese

An example run of the program (numbers in bold are typed in by the user)

Enter your weight in (kg): 75

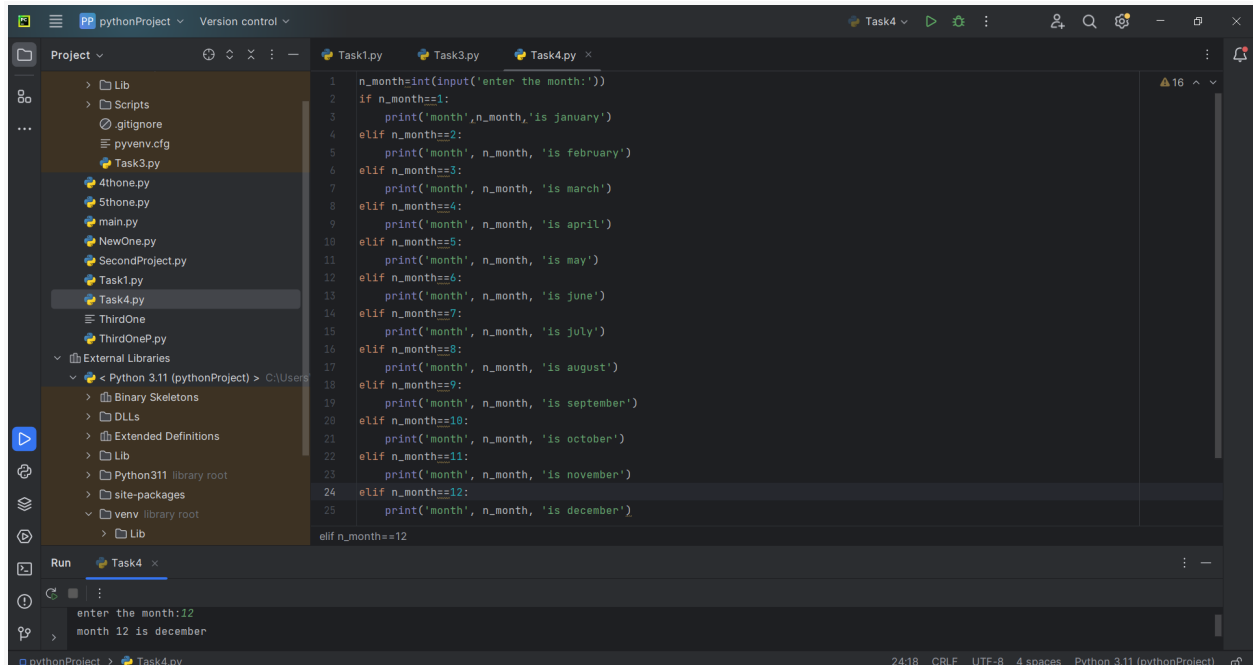
Enter your height in (m): 1.70

Your BMI is: 25.95

You are in the “overweight” range.

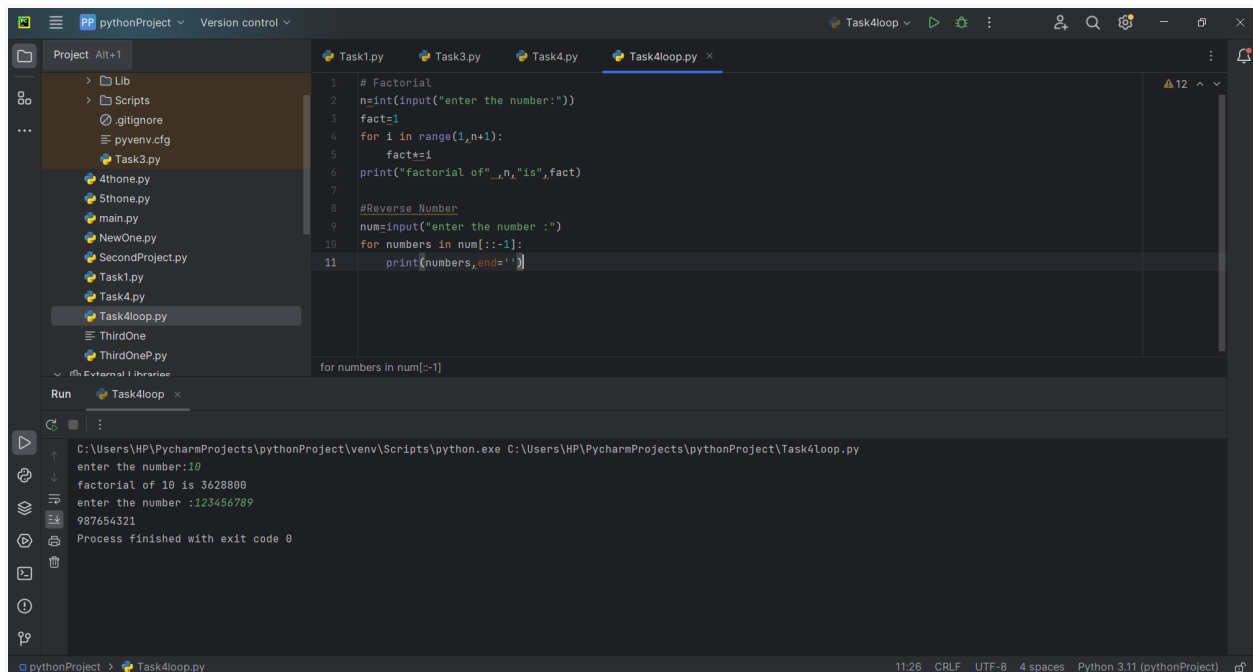
LOOPS:

1. Find the factorial of a given number using loops(note the number is received from the user)
2. Finding the multiples of a number using a loops.
3. Write a program to find the reverse of a number using loops. (edited)



The screenshot shows the PyCharm IDE with a project named 'pythonProject'. The file explorer on the left shows a directory structure with files like 'Task3.py', 'Task4.py', and 'Task4loop.py'. The main editor window displays the code for 'Task4.py', which is a program that takes a month number as input and prints the corresponding month name. The code uses a loop to iterate through the months of the year. The Run console at the bottom shows the output of the program: 'enter the month:12' and 'month 12 is december'.

```
1 n_month=int(input('enter the month:'))
2 if n_month==1:
3     print('month',n_month,'is january')
4 elif n_month==2:
5     print('month', n_month, 'is february')
6 elif n_month==3:
7     print('month', n_month, 'is march')
8 elif n_month==4:
9     print('month', n_month, 'is april')
10 elif n_month==5:
11     print('month', n_month, 'is may')
12 elif n_month==6:
13     print('month', n_month, 'is june')
14 elif n_month==7:
15     print('month', n_month, 'is july')
16 elif n_month==8:
17     print('month', n_month, 'is august')
18 elif n_month==9:
19     print('month', n_month, 'is september')
20 elif n_month==10:
21     print('month', n_month, 'is october')
22 elif n_month==11:
23     print('month', n_month, 'is november')
24 elif n_month==12:
25     print('month', n_month, 'is december')
elif n_month==12
```



The screenshot shows the PyCharm IDE with a project named 'pythonProject'. The file explorer on the left shows a directory structure with files like 'Task3.py', 'Task4.py', and 'Task4loop.py'. The main editor window displays the code for 'Task4loop.py', which is a program that takes a number as input and prints its factorial and reverse. The code uses loops to calculate the factorial and reverse the number. The Run console at the bottom shows the output of the program: 'enter the number:10', 'factorial of 10 is 3628800', 'enter the number :123456789', '987654321', and 'Process finished with exit code 0'.

```
1 # Factorial
2 n=int(input("enter the number:"))
3 fact=1
4 for i in range(1,n+1):
5     fact*=i
6 print("factorial of",n,"is",fact)
7
8 #Reverse Number
9 num=input("enter the number :")
10 for numbers in num[::-1]:
11     print(numbers,end='')
for numbers in num[::-1]
```

This screenshot shows the PyCharm IDE with a Python project named 'pythonProject'. The file explorer on the left lists various files, including 'Task4.py'. The main editor window displays the code for 'Task4.py', which includes a function to print the month name based on a number and a function to check if a number is even or odd. The Run console at the bottom shows the execution of the program with the following output:

```
C:\Users\HP\PycharmProjects\pythonProject\venv\Scripts\python.exe C:\Users\HP\PycharmProjects\pythonProject\Task4.py
enter the month:12
month 12 is december
enter a number :14
14 is "even"
Process finished with exit code 0
```

This screenshot shows the PyCharm IDE with the same 'pythonProject'. The file explorer on the left lists various files, including 'Task4.py'. The main editor window displays the code for 'Task4.py', which includes a function to print the ticket cost based on age. The Run console at the bottom shows the execution of the program with the following output:

```
C:\Users\HP\PycharmProjects\pythonProject\venv\Scripts\python.exe C:\Users\HP\PycharmProjects\pythonProject\Task4.py
enter the month:12
month 12 is december
enter a number :4
4 is "even"
Enter your age:50
your ticket costs pounds 6
Process finished with exit code 0
```

The screenshot shows the PyCharm IDE with a project named 'pythonProject'. The file explorer on the left shows a directory structure with files like 'Task3.py', 'Task4.py', and 'Task4loop.py'. The main editor window displays 'Task4.py' with the following code:

```
42
43 # Exercise 4 bodymass index
44 weight=float(input("enter your weight in (kg):"))
45 height=float(input("enter your height in (m):"))
46 BMI=weight/height**2
47 rounded_BMI=round(BMI,2)
48 print("your BMI is",rounded_BMI)
49 if rounded_BMI<18.5:
50     print("you are in the \"underweight\" range")
51 elif 18.5<rounded_BMI<24.9:
52     print("you are in the \"normal\" range")
53 elif 25<rounded_BMI<29.9:
54     print("you are in the \"overweight\" range")
55 elif rounded_BMI>=30:
56     print("you are in the \"obese\" range")
57
```

The Run window at the bottom shows the execution of 'Task4.py' with the following output:

```
C:\Users\HP\PycharmProjects\pythonProject\venv\Scripts\python.exe C:\Users\HP\PycharmProjects\pythonProject\Task4.py
enter your weight in (kg):74
enter your height in (m):1.75
your BMI is 24.16
you are in the "normal" range
Process finished with exit code 0
```

The screenshot shows the PyCharm IDE with a project named 'pythonProject'. The file explorer on the left shows a directory structure with files like 'Task3.py', 'Task4.py', and 'Task4loop.py'. The main editor window displays 'Task4loop.py' with the following code:

```
13
14 #Multiples
15 n=int(input("enter the number:"))
16 for i in range(1,11,1):
17     result=i*n
18     print(i,"*",n,"=",result)
19
20
21
22
23
24
25
```

The Run window at the bottom shows the execution of 'Task4loop.py' with the following output:

```
C:\Users\HP\PycharmProjects\pythonProject\venv\Scripts\python.exe C:\Users\HP\PycharmProjects\pythonProject\Task4loop.py
enter the number:10
1 * 10 = 10
2 * 10 = 20
3 * 10 = 30
4 * 10 = 40
5 * 10 = 50
6 * 10 = 60
7 * 10 = 70
8 * 10 = 80
9 * 10 = 90
10 * 10 = 100
Process finished with exit code 0
```

Task 5The task for class conducted on 11-09-2023

Break and Continue & Nested Loops

1 Name your file: TrianglePattern.py

Write a program that display a triangle made up of asterisk symbols.

An example runs of the program

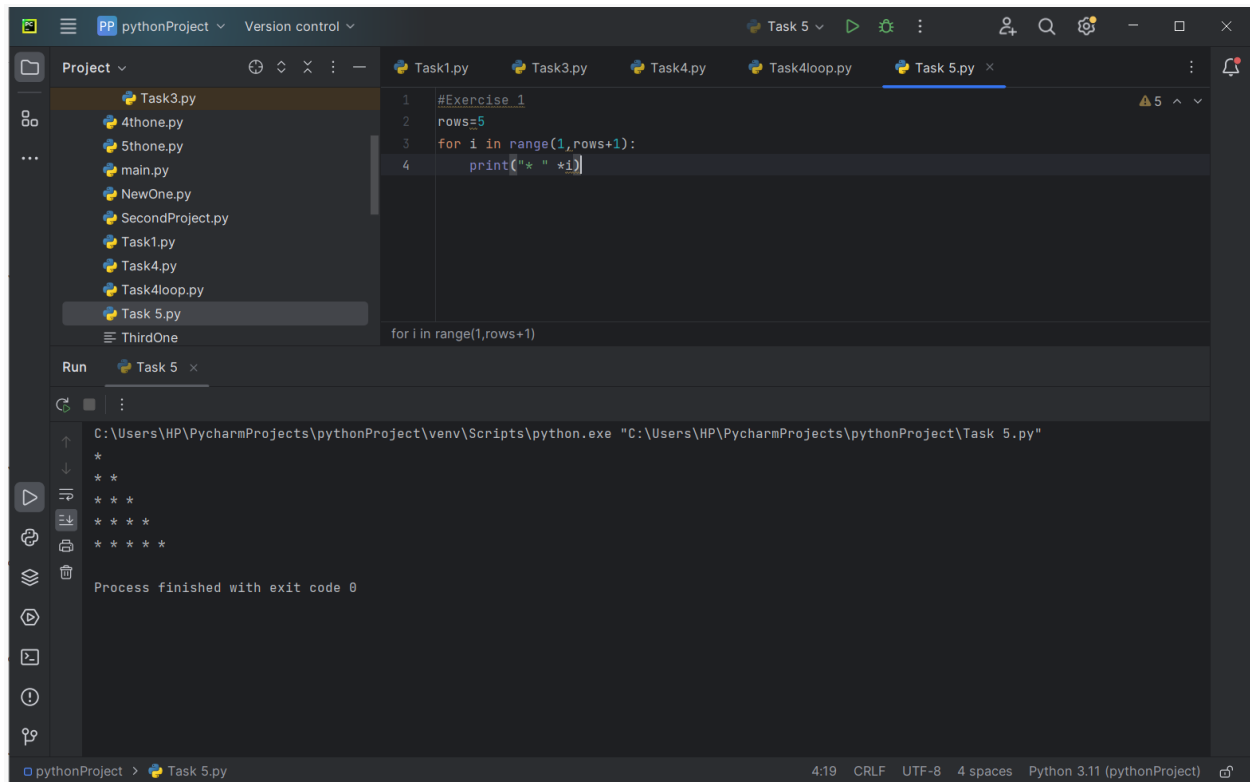
```
*
* *
* * *
```



```
* * * * *  
* * * * *
```

2 Write a program that prompts the user for a positive integer, and then outputs a list of all positive integers that are less than or equal to the input value. However, if the input value is less than 1, the program should print "Invalid Input" and break.

3 Write a program that calculates the sum of the first n natural numbers. The program should prompt the user for a positive integer n, and then output the sum of the first n natural numbers. However, if the input value is less than 1, the program should print "Invalid Input" and break.



The screenshot shows the PyCharm IDE interface. The top toolbar includes icons for running and debugging. The 'Project' view on the left lists files: Task3.py, 4thone.py, 5thone.py, main.py, NewOne.py, SecondProject.py, Task1.py, Task4.py, Task4loop.py, Task 5.py, and ThirdOne. The 'Task 5.py' file is open in the editor, showing the following code:

```
1 #Exercise 1  
2 rows=5  
3 for i in range(1,rows+1):  
4     print("*" * i)
```

The 'Run' window at the bottom shows the command executed: `C:\Users\HP\PycharmProjects\pythonProject\venv\Scripts\python.exe "C:\Users\HP\PycharmProjects\pythonProject\Task 5.py"`. The output is:

```
*  
* *  
* * *  
* * * *  
* * * * *
```

Below the output, it states: "Process finished with exit code 0". The status bar at the bottom indicates the file encoding is UTF-8, uses 4 spaces for indentation, and is running Python 3.11.

The screenshot shows the PyCharm IDE with a project named 'pythonProject'. The file explorer on the left lists several files, including 'Task 5.py'. The main editor window displays the code for 'Task 5.py', which is a Python script for calculating the sum of integers from 1 to n. The code includes a comment '#Exercise 2 sumnatural', a function definition 'def sumnatural(n):', and a loop that calculates the sum. The Run window at the bottom shows the execution of the script, with the input '10' and the output '55'.

```
5
6
7 #Exercise 2 sumnatural
8 n=int(input("enter a positive integer :"))
9 total=0
10 if n<1:
11     print("invalid input")
12 else:
13     for i in range(1,n+1):
14         total+=i
15     print(total)
16
else:
    for i in range(1,n+1)
```

Run Task 5

C:\Users\HP\PycharmProjects\pythonProject\venv\Scripts\python.exe "C:\Users\HP\PycharmProjects\pythonProject\Task 5.py"

enter a positive integer :10

55

Process finished with exit code 0

The screenshot shows the PyCharm IDE with the same project 'pythonProject'. The file explorer on the left lists several files, including 'Task 5.py'. The main editor window displays the code for 'Task 5.py', which is a Python script for calculating the sum of integers from n down to 1. The code includes a comment '#Exercise3', a function definition 'def sumnatural(n):', and a loop that calculates the sum. The Run window at the bottom shows the execution of the script, with the input '10' and the output '10 9 8 7 6 5 4 3 2 1'.

```
16
17
18 #Exercise3
19 n=int(input("enter a positive integer:"))
20 if n<1:
21     print("invalid input")
22 else:
23     for i in range(n, 0, -1):
24         print(i)
25
```

Run Task 5

C:\Users\HP\PycharmProjects\pythonProject\venv\Scripts\python.exe "C:\Users\HP\PycharmProjects\pythonProject\Task 5.py"

enter a positive integer:10

10

9

8

7

6

5

4

3

2

1

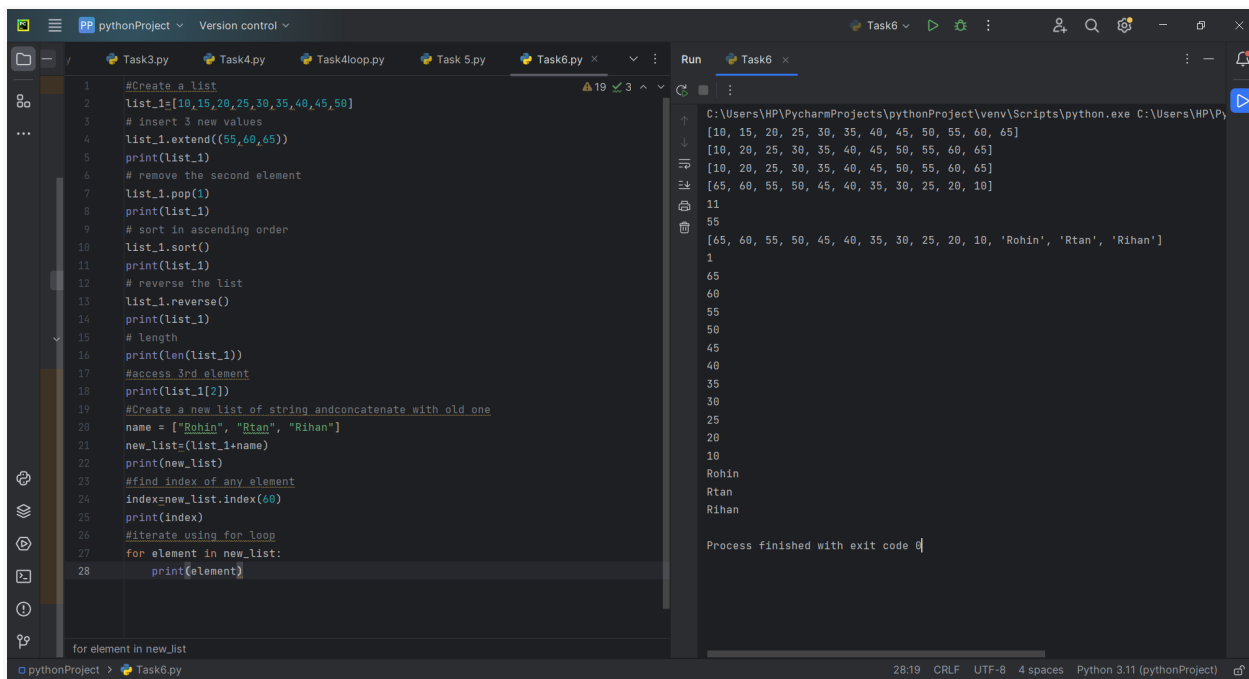
Process finished with exit code 0

Task 6The task for class conducted on 18-09-2023

List

Create a list and perform the following:

- 1 Insert 3 new values into the list and print the updated list.
- 2 Remove the second element from the list and print the updated list.
- 3 Sort the list in ascending order and print the sorted list.
- 4 Reverse the list and print the reversed list.
- 5 Find the length of the list and print the length.
- 6 Access the third element in the list and print the element.
- 7 Create a new list of strings and concatenate the two lists into a third list.
- 8 Find the index of a specific element in the list and print the index.
- 9 Use a for loop to print each element in the list.



The screenshot shows the PyCharm IDE with a Python script named `Task6.py` and its execution output in the Run console.

```
1 #Create a list
2 list_1=[10,15,20,25,30,35,40,45,50]
3 # insert 3 new values
4 list_1.extend((55,60,65))
5 print(list_1)
6 # remove the second element
7 list_1.pop(1)
8 print(list_1)
9 # sort in ascending order
10 list_1.sort()
11 print(list_1)
12 # reverse the list
13 list_1.reverse()
14 print(list_1)
15 # length
16 print(len(list_1))
17 #access 3rd element
18 print(list_1[2])
19 #Create a new list of string and concatenate with old one
20 name = ["Rohin", "Rtan", "Rihan"]
21 new_list=(list_1+name)
22 print(new_list)
23 #find index of any element
24 index=new_list.index(60)
25 print(index)
26 #iterate using for loop
27 for element in new_list:
28     print(element)
```

The Run console output shows the following sequence of lists and values:

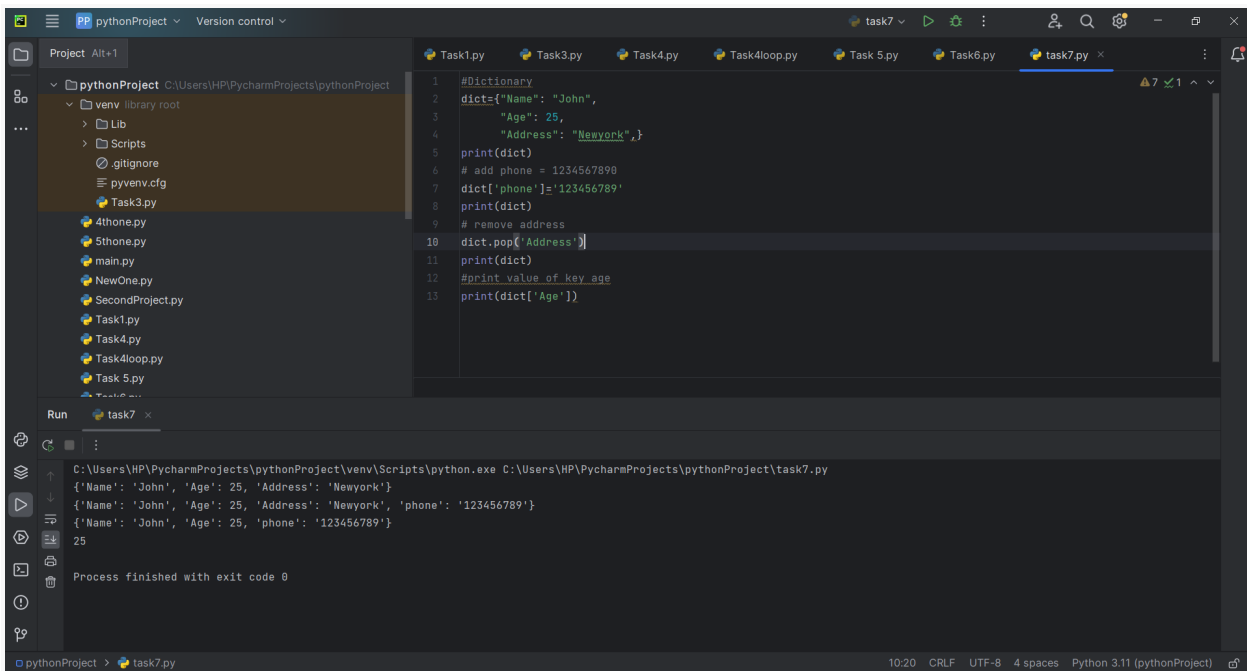
```
[10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65]
[10, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65]
[10, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65]
[65, 60, 55, 50, 45, 40, 35, 30, 25, 20, 10]
11
55
[65, 60, 55, 50, 45, 40, 35, 30, 25, 20, 10, 'Rohin', 'Rtan', 'Rihan']
1
65
60
55
50
45
40
35
30
25
20
10
Rohin
Rtan
Rihan
```

Process finished with exit code 0

Task 7 The task for class conducted on 22-09-2023

Dictionary

- 1 Create a dictionary with keys 'name', 'age', and 'address' and values 'John', 25, and 'New York' respectively.
2. Add a new key-value pair to the dictionary created in Q1 with key 'phone' and value '1234567890'.
3. Remove the key 'address' from the dictionary created in Q1.
4. Print the value of the key 'age' from the dictionary created in Q1.



The screenshot shows the PyCharm IDE interface. The left sidebar displays the project structure for 'pythonProject', including a 'venv' directory and various Python files like 'Task1.py' through 'Task5.py'. The main editor window is open to 'task7.py', which contains the following Python code:

```
1 #Dictionary
2 dict={"Name": "John",
3       "Age": 25,
4       "Address": "Newyork",}
5 print(dict)
6 # add phone = 1234567890
7 dict['phone']='123456789'
8 print(dict)
9 # remove address
10 dict.pop('Address')
11 print(dict)
12 #print value of key age
13 print(dict['Age'])
```

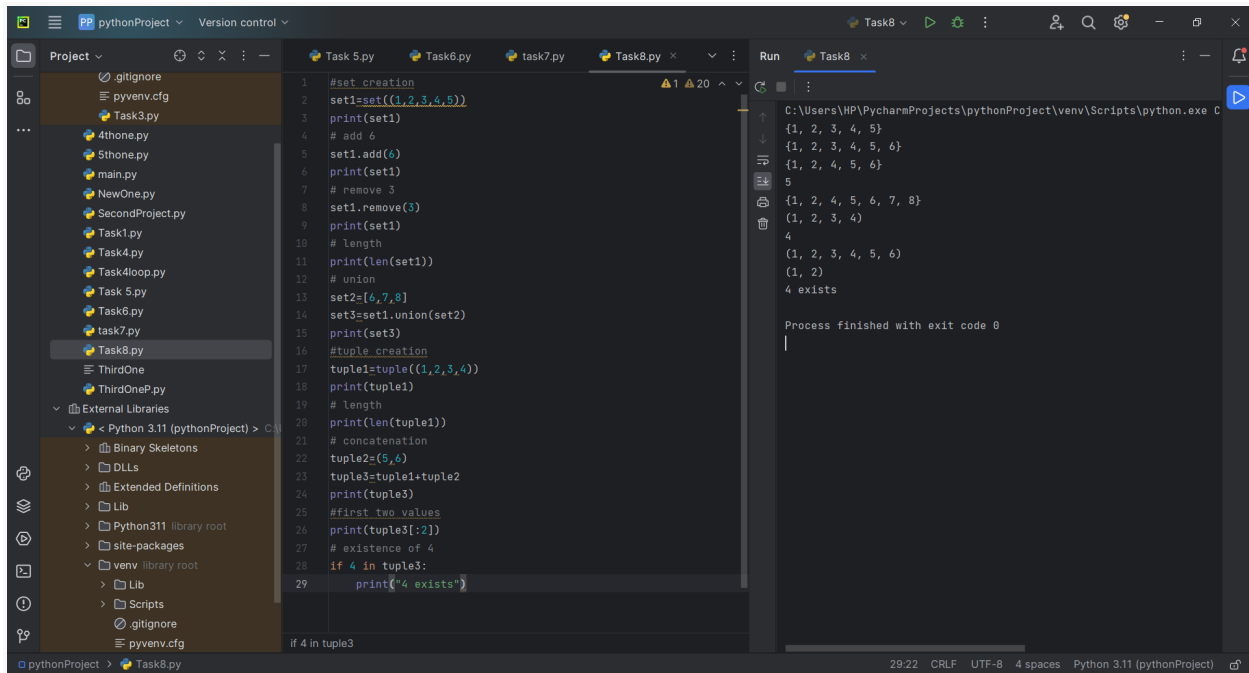
Below the editor, the 'Run' tab shows the execution output for 'task7.py'. The output displays the dictionary at three stages: initial creation, after adding the 'phone' key, and after removing the 'Address' key. The final output is '25', which is the value associated with the 'Age' key. The status bar at the bottom indicates the process finished with exit code 0.

Task8 Task for class conducted on 25-09-2023

Sets, Tuples

- 1 Create a set with values 1, 2, 3, 4, and 5.
- 2 Add the value 6 to the set created in Question 1.
- 3 Remove the value 3 from the set created in Question 1.
- 4 Print the length of the set created in Question 1.
- 5 Create a new set by union of the set created in Question 1 with another set {6, 7, 8}.
- 6 Create a tuple with values 1, 2, 3, and 4
- 7 Print the length of the tuple created in Question 6.

- 8 Create a new tuple by concatenating the tuple from Question 6 with another tuple (5, 6).
- 9 Print the first two values of the tuple created in Question 8.
- 10 Check if the value 4 exists in the tuple created in Question 1.



The screenshot shows the PyCharm IDE interface. The left sidebar displays the project structure with files like `Task3.py`, `Task4.py`, `Task5.py`, `Task6.py`, `Task7.py`, and `Task8.py`. The main editor window shows the code for `Task8.py`, which includes set creation, modification, and tuple operations. The right sidebar shows the output of the program, displaying the execution of the code and the final result.

```
1 #set creation
2 set1=set((1,2,3,4,5))
3 print(set1)
4 # add 6
5 set1.add(6)
6 print(set1)
7 # remove 3
8 set1.remove(3)
9 print(set1)
10 # length
11 print(len(set1))
12 # union
13 set2=[6,7,8]
14 set3=set1.union(set2)
15 print(set3)
16 #tuple creation
17 tuple1=tuple((1,2,3,4))
18 print(tuple1)
19 # length
20 print(len(tuple1))
21 # concatenation
22 tuple2=(5,6)
23 tuple3=tuple1+tuple2
24 print(tuple3)
25 #first two values
26 print(tuple3[:2])
27 # existence of 4
28 if 4 in tuple3:
29     print("4 exists")
```

Output:

```
{1, 2, 3, 4, 5}
{1, 2, 3, 4, 5, 6}
{1, 2, 4, 5, 6}
5
{1, 2, 4, 5, 6, 7, 8}
(1, 2, 3, 4)
4
(1, 2, 3, 4, 5, 6)
(1, 2)
4 exists
Process finished with exit code 0
```

Task 9Task for class on 06-10-2023:

Functions:

Exercises 1

Write a Python program to count the number of strings where the string length is 2 or more and the first and last character are same from a given list of strings

Sample List : ['abc', 'xyz', 'aba', '1221']

Expected Result : 2

Exercises 2

Write a Python program to multiply all the items in a list

Exercise 3

Create dictionary from a list where the keys are the elements of the list and value of the dictionary is result after dividing the element by 3

The screenshot shows the PyCharm IDE with a project named 'pythonProject'. The file explorer on the left shows a directory structure with files like '3.py', 'task7.py', 'Task8.py', and 'Task9.py'. The main editor window displays the code for 'Task9.py', which contains three exercises:

```
1 #Exercise 1
2 def string_func(input_list):
3     count=0
4     for s in (input_list):
5         if len(input_list)>=2 and s[0]==s[-1]:
6             count+=1
7         print(count)
8
9 list_1=['abc','xyz','aba','1221']
10 string_func(list_1)
11
12 #Exercise 2 multiply all the items in the list
13
14 usage
15 def list_multiply(input_list):
16     multiply=1
17     for num in input_list:
18         multiply=multiply*num
19         print(multiply)
20     list_1=[1,2,3,4,5]
21     list_multiply(list_1)
22
23 #Exercise 3
24 list_1=[1,2,3,4,5]
25 dict_1={num:num/3 for num in list_1}
26 print(dict_1)
```

The Run window on the right shows the execution output for 'Task9.py':

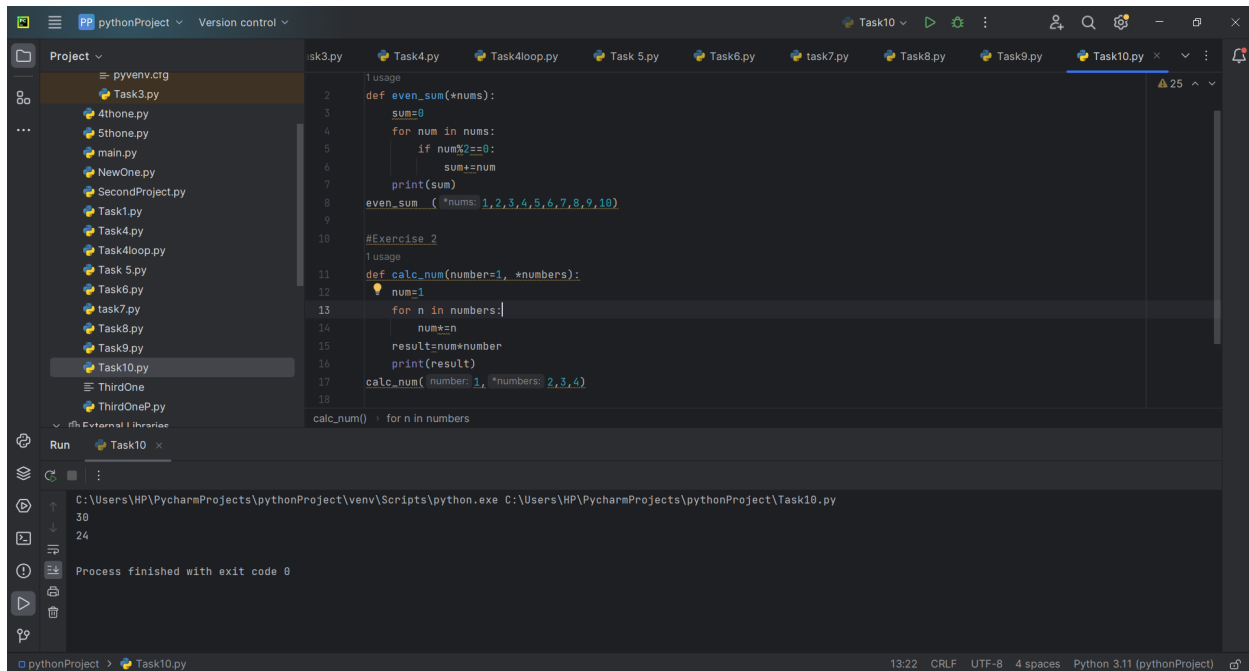
```
0
0
1
2
1
2
6
24
120
{1: 0.3333333333333333, 2: 0.6666666666666666, 3: 1.0, 4: 1.3333333333333333, 5: 1.6666666666666667}
```

Process finished with exit code 0

Task 10 Task for class conducted on 09-10-2023

Functions:

- 1 Create a function that takes in an arbitrary number of arguments and prints the sum of all the even numbers. If no even numbers are provided, the function should print 0.
- 2 Create a function that takes in two required arguments, one optional keyword argument with a default value of 1, and one arbitrary argument. The function should print the product of the two required arguments, multiplied by the optional keyword argument, multiplied by the product of all the values in the arbitrary argument.



Task 11 Task for class conducted on 13-10-2023:

Functions:

Write a function to implement calculator

Sample run:

Press 1 for addition

Press 2 for subtraction

Press 3 for Multiplication

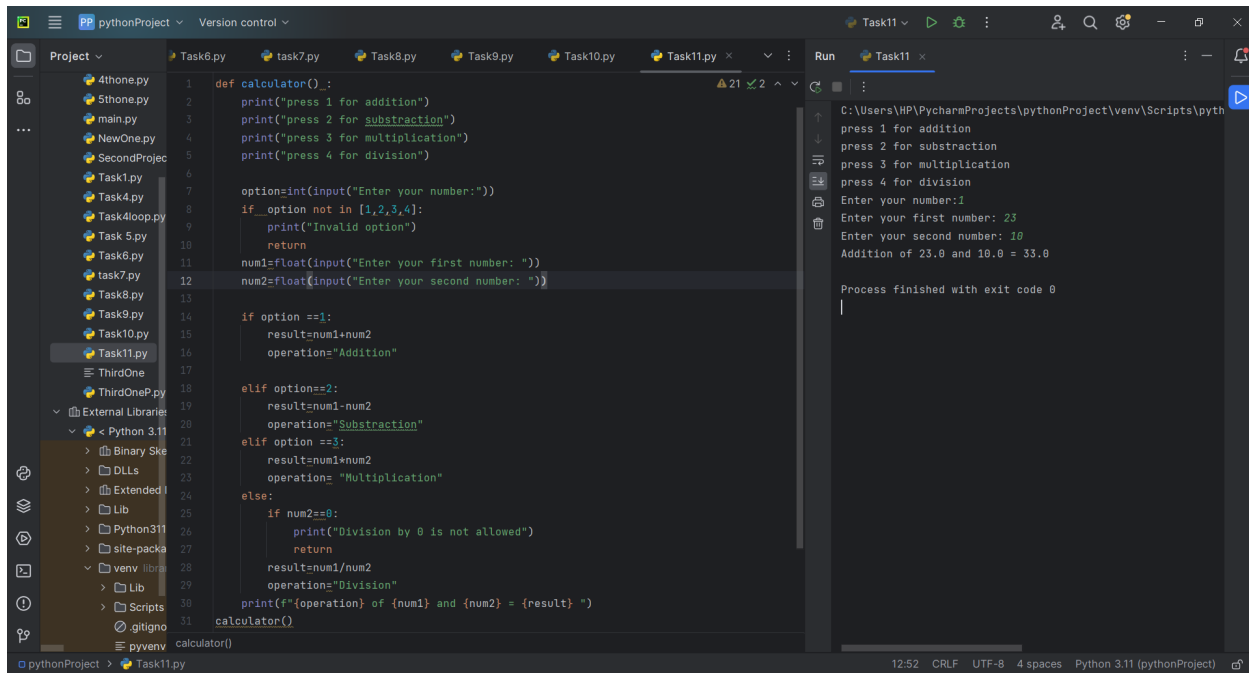
Press 4 for Division

Enter your option: 1

Enter your first number: 23

Enter your second number: 10

Addition of 23 and 10 is : 33



Task 12

Task for class on 16-10-2023:

Lambda...reduce()

- 1 Write a lambda function that takes in a parameter and returns True if the parameter is even and False otherwise.
- 2 Write a Python program to create a lambda function that adds 15 to a given number passed in as an argument, also create a lambda function that multiplies argument x with argument y and prints the result.
- 3 Write a Python program to evaluate a given mathematical expression using the eval() function.
 expression = "3 * 5 + 2"
- 4 Write a Python program to evaluate a given logical expression using the eval() function.
 expression = "5 > 2 and 3 < 4"
- 5 Write a Python program to evaluate a given Python expression using the eval() function.
 expression = "[x * 2 for x in range(5)]"
- 6 Write a Python program to filter out the vowels from a given list of characters using the filter() function.
- 7 Write a Python program to filter out the prime numbers from a given list of integers using the filter() function.
- 8 Write a Python program to filter out the strings with length greater than 5 from a given list of strings using the filter() function


```
1 # Exercise 1
2 even=lambda x: x%2==0
3 print(even(4))
4 print(even(7))
5
6 # Exercise 2
7 add=lambda y:15+y
8 print(add(30))
9 multiply=lambda x,y:x*y
10 print(add(5))
11 print(multiply(3, 4))
12
13 #Exercise 3
14 expression="3*5+2"
15 print(eval(expression))
16
17 #Exercise 4
18 expression2="5>2 and 3<4"
19 print(eval(expression2))
20
21 #Exercise 5
22 expression3="[x*2 for x in range(5)]"
23 print(eval(expression3))
24
25 #Exercise 6
26 characters_list=['a','b','c','d','e','i','o','u']
27 vowels=list(filter(lambda a:a in 'aeiou',characters_list)))
28 print(vowels)
29
30 #Exercise 7
31
```

```
29
30 #Exercise 7
31
32 def prime_num(nums):
33     if nums<=1:
34         return False
35     if nums<=3:
36         return True
37     if nums%2==0 or nums%3==0:
38         return False
39     i=5
40     while i<=nums:
41         if nums%i==0 or nums%(i+2)==0:
42             return False
43         i+=6
44     return True
45 number_list = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
46 print(list(filter(prime_num,number_list)))
47
48 #Exercise 8
49 strings=['orange','apple','jackfruit','reeta','fahad']
50 lengths_list(filter(lambda x: len(x)>5,strings))
51 print(lengths)
52
53
```

Task 13 Task for class on 20-10-2023:

Generators and Decorators:

Exercise 1:

Write a generator function that takes a string and yields each word in the string

Exercise 2:

Write a generator function that takes a list of integers and yields only the even numbers

Exercise 3:

Write a generator function that takes a list of strings and yields only the strings that start with a vowel.

Exercise 4:

Write a decorator function that logs the arguments and return value of a function.

The screenshot shows a Python IDE with a file explorer on the left, a code editor in the center, and a run console on the right. The code editor displays the following Python code:

```
1 #Exercise 1
2 usage
3 def word_str(text):
4     words = text.split()
5     for word in words:
6         yield word
7
8 text="This is a generator function"
9 result=word_str(text)
10 for word in result:
11     print(word)
12
13 # Exercise 2
14 usage
15 def even(num):
16     for n in num:
17         if n%2==0:
18             yield n
19
20 num=[2,4,6,8,7,9,11,12,13,15]
21 for even_num in even(num):
22     print(even_num)
23
24 # Exercise 3
25 usage
26 def vowel_str(str):
27     vowels='aeiou'
28     for words in str:
29         if words[0] in vowels:
30             yield words
31
32 str='ant','book','chain','inkpot','pain','jungle','laptop'
33 result=vowel_str(str)
34 for word in result:
35     print(word)
```

The run console on the right shows the output of the code:

```
C:\Users\HP\PycharmProjects\pythonProject\venv\Scripts\pyth
This
is
a
generator
function
2
4
6
8
12
ant
inkpot

Process finished with exit code 0
```

The screenshot shows a Python IDE with a file explorer on the left, a code editor in the center, and a run console on the right. The code editor displays the following Python code:

```
1 # Exercise 4 decorators
2 usage
3 def log_function(func):
4     def wrapper(*args, **kwargs):
5         print(f'Arguments :args={args}, kwargs={kwargs}')
6         result = func(*args, **kwargs)
7         print(f'Function add_num return value:{result}')
8         return result
9     return wrapper
10
11 @log_function
12 def add_num(a, b):
13     return a + b
14
15 result = add_num(10, 20)
```

The run console on the right shows the output of the code:

```
C:\Users\HP\PycharmProjects\pythonProject\venv\Scripts\pyth
Arguments :args=(10, 20), kwargs={}
Function add_num return value:30

Process finished with exit code 0
```