A screenshot of a computer

Description automatically generated

Yes changing the value of a variable which has benn assigned a string previously to int is possible.



A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Operations between string and int is not allowed in python.

Addition between 2 string leads to concatenation.

And other operations between strings also not allowed.

Operations between int and float are allowed. Implicit type conversion takes places in which small data type gets converted to larger one to avoid data loss.

To perform operations between numeric strings like “123” we need to do explicit type conversion.

value = “123”

num = int(value) # num = 123



A screenshot of a computer

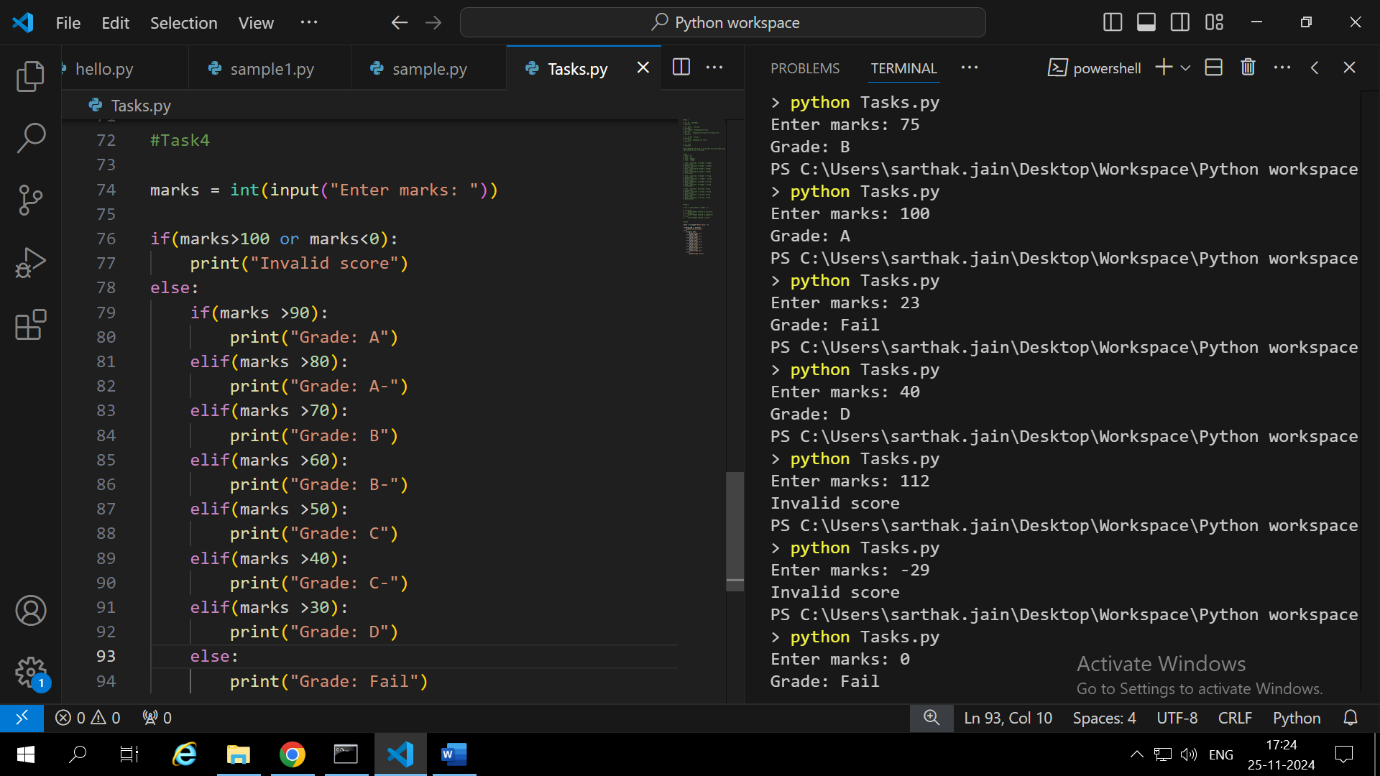
Description automatically generated

A screenshot of a computer

Description automatically generated

If we enter a string in place of numeric value “TypeError” occurs. Since python can’t convert string to int.





We can add an additional if condition to check for invalid cases and then in else we can write the logic for valid cases.



A screenshot of a computer

Description automatically generated

A screenshot of a computer

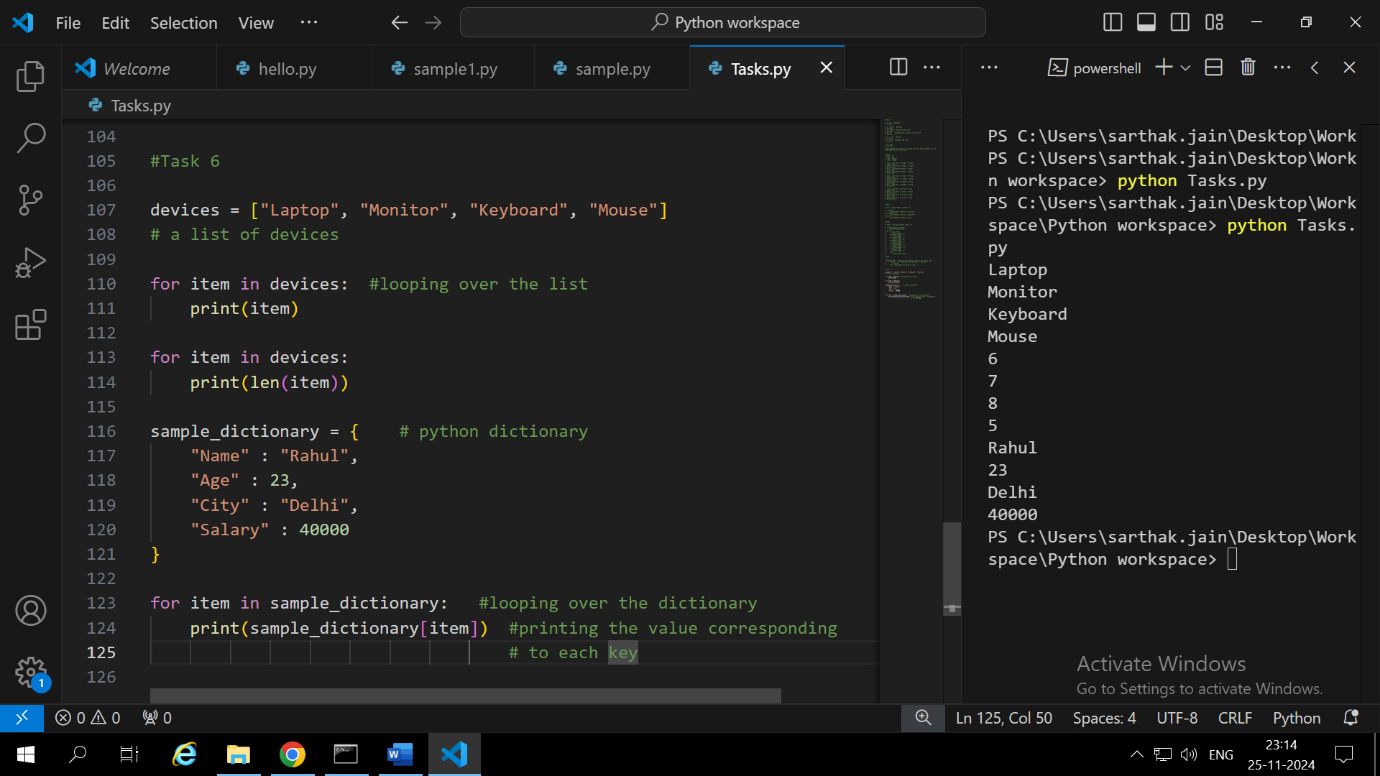
Description automatically generated

To print the even numbers we have added an additional condition :

If(x%2 == 0)

If the condition is never updated in the while loop then the loop will run infinitely causing memory overflow.



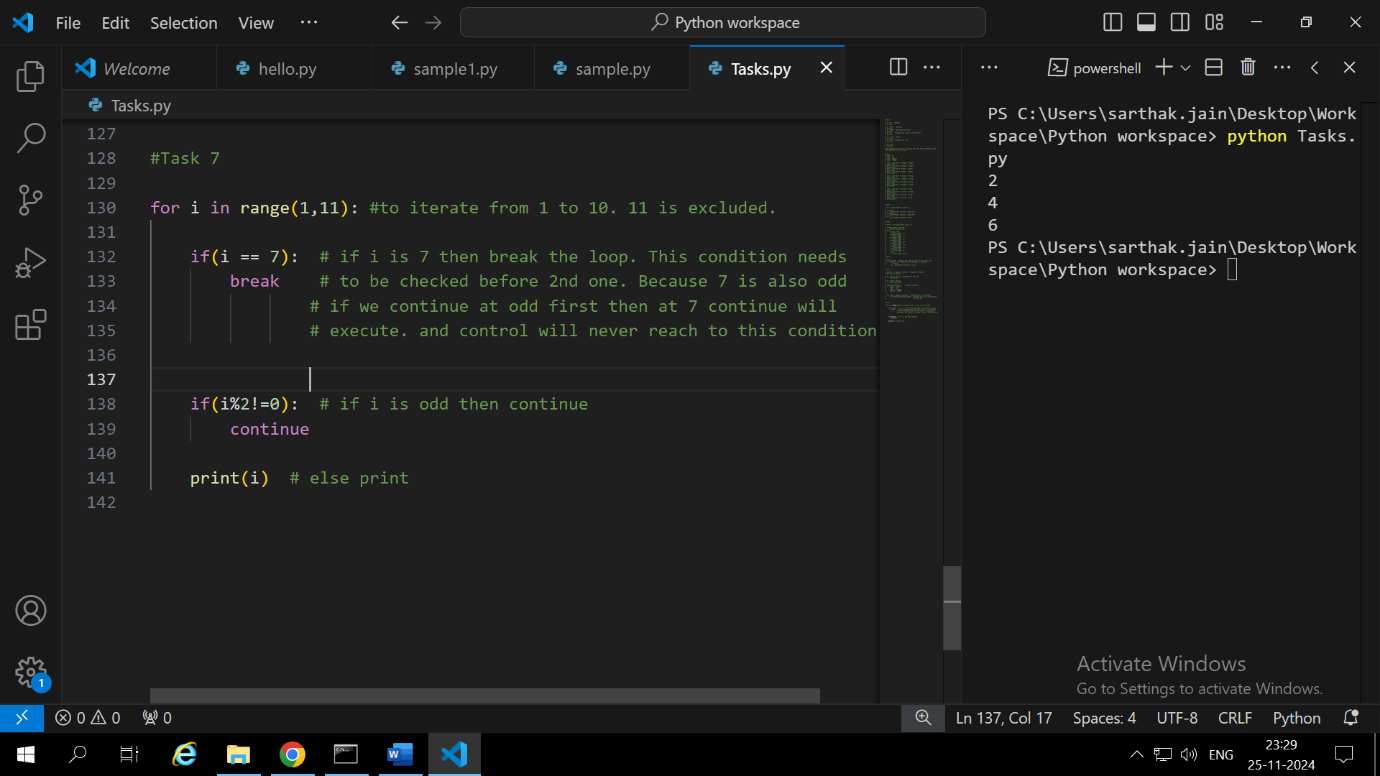


For looping over the dictionary use for loop:

for item in dict: #item refers to each key in the dict

print(dict[key]) #for accessing the value corresponding to the key





Using break in for loop makes the control to exit the loop.

And continue is used when we want to execute anything at a certain point. The control again goes to top when it encounters continue.



A screenshot of a computer

Description automatically generated

Type casting means converting data from one type to another.

In python it occurs in 2 ways: implicit and explicit

Implicit when type conversion occurs automatically without user input

Explicit when type conversion requires manual command by user.

Python implicitly converts data from smaller data type to bigger data type to avoid data loss since operation between 2 different data types is not possible. They both need to be of the same type.

When we explicitly convert float to int then decimal value gets lost. Only the integer part remains.



A screenshot of a computer

Description automatically generated

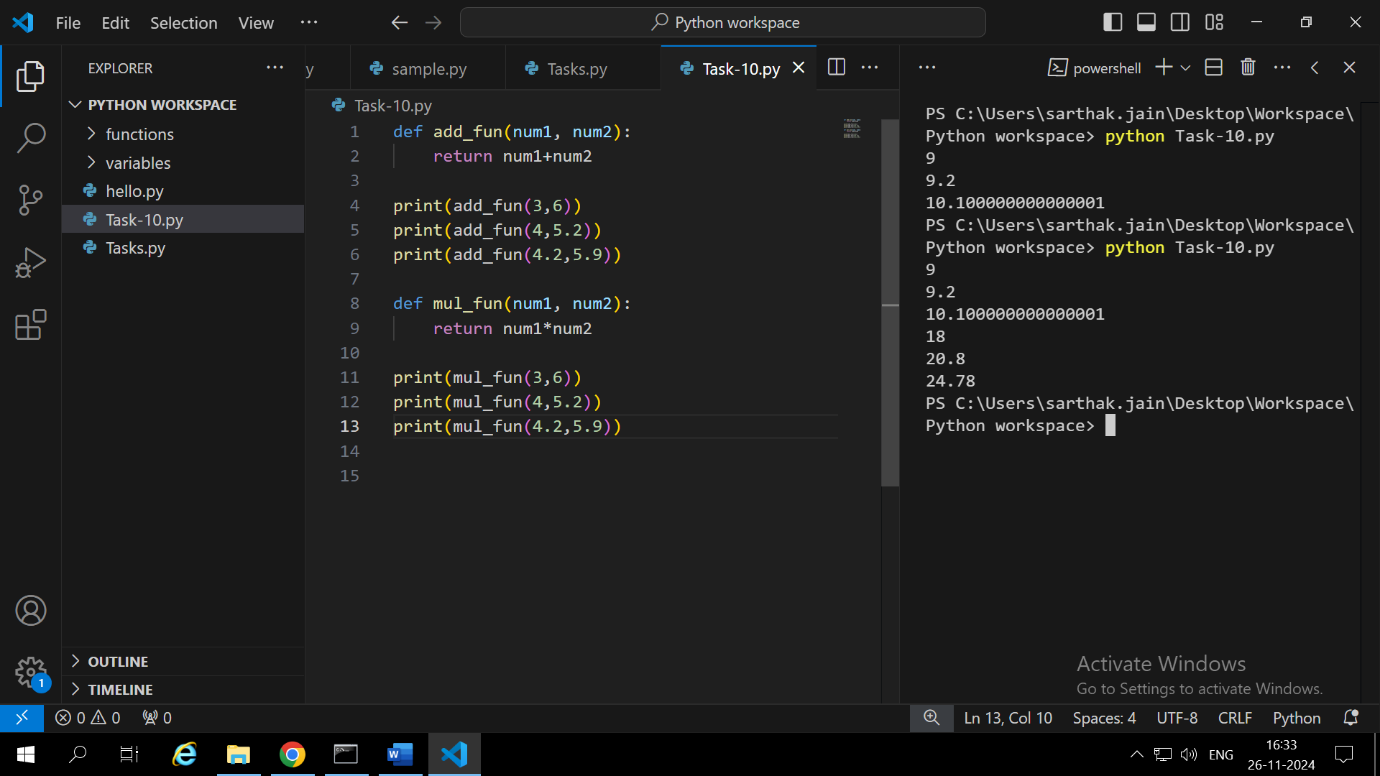
Try a block of code in try: . If error occurs it will be handles by “except”. If try-except is not used the error will lead to failure of the code. If no error occurs then the else block will run.

If the user enters numeric value then error will occur and code will fail. To handle this we need to put the input statement also in the try block.

A screenshot of a computer

Description automatically generated





Just place ‘\*’ in place of ‘+’ to make the function return product of 2 numbers