

Rushabh Shah

Dr. Muhammad Zubair Khan

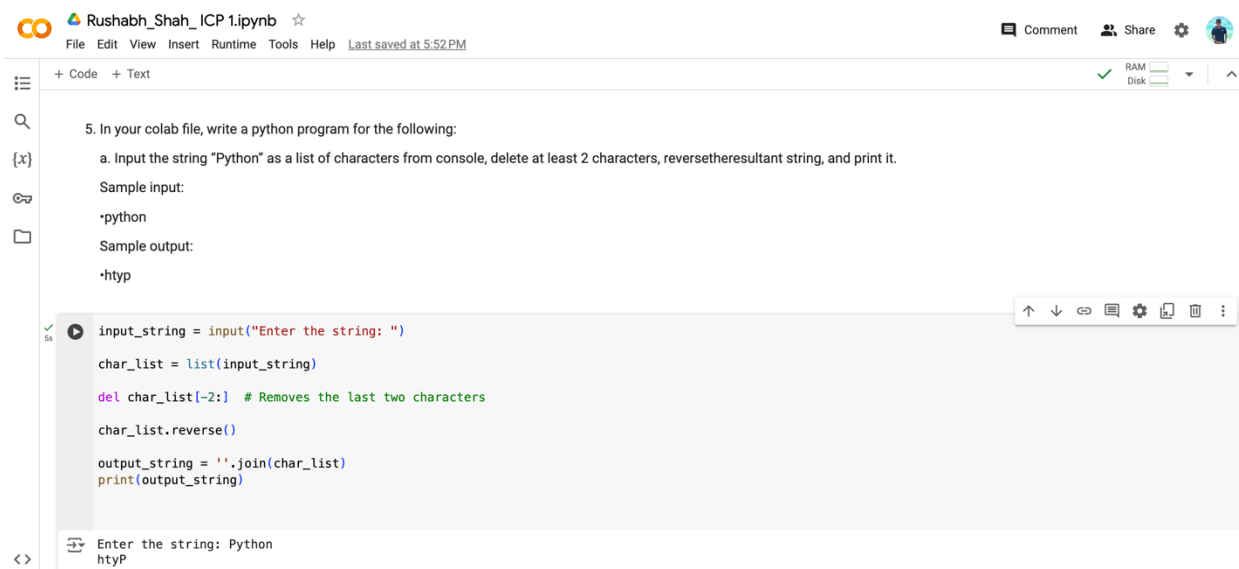
DSA 4620

August 23, 2024

5) Input the string “Python” as a list of characters from console, delete at least 2 characters, reverse the resultant string, and print it.

Sample input: •python

Sample output: •htyp



The screenshot shows a Jupyter Notebook interface with a file named "Rushabh_Shah_ICP 1.ipynb". The notebook contains a single code cell with the following Python code:

```
input_string = input("Enter the string: ")
char_list = list(input_string)
del char_list[-2:] # Removes the last two characters
char_list.reverse()
output_string = ''.join(char_list)
print(output_string)
```

Below the code cell, the input and output are shown:

```
Enter the string: Python
htyp
```

b. Take two numbers from user and perform at least 4 arithmetic operations on them.

The screenshot shows a Jupyter Notebook titled "Rushabh_Shah_ICP 1.ipynb". The code is as follows:

```
# Step 1:
num1 = float(input("Enter the first number: "))
num2 = float(input("Enter the second number: "))

# Step 2:
addition = num1 + num2
subtraction = num1 - num2
multiplication = num1 * num2
division = num1 / num2 if num2 != 0 else "undefined (division by zero)"

# Step 3:
print(f"Addition: {num1} + {num2} = {addition}")
print(f"Subtraction: {num1} - {num2} = {subtraction}")
print(f"Multiplication: {num1} * {num2} = {multiplication}")
print(f"Division: {num1} / {num2} = {division}")
```

The output of the program is:

```
Enter the first number: 7
Enter the second number: 10
Addition: 7.0 + 10.0 = 17.0
Subtraction: 7.0 - 10.0 = -3.0
Multiplication: 7.0 * 10.0 = 70.0
Division: 7.0 / 10.0 = 0.7
```

6. Write a program that accepts a sentence and replace each occurrence of 'python' with 'pythons'.

Sample input:

• I love playing with python

• Sample output: • I love playing with pythons

The screenshot shows a Jupyter Notebook titled "Rushabh_Shah_ICP 1.ipynb". The code is as follows:

```
def replace_python(sentence):
    updated_sentence = sentence.replace('python', 'pythons')
    return updated_sentence

# Enter a sentence
input_sentence = input("Enter a sentence: ")

# user's input and store the result
result = replace_python(input_sentence)

# Print the updated sentence
print("Updated sentence:", result)
```

The output of the program is:

```
Enter a sentence: I love playing with python
Updated sentence: I love playing with pythons
```

7. Use the if statement conditions to write a program to print the letter grade based on an input class score. Use the grading scheme we are using in this class.



The screenshot shows a Jupyter Notebook titled "Rushabh_Shah_ICP 1.ipynb". The code in the cell is as follows:

```
score = float(input("Enter the class score (0-100): "))

# letter grade using if-elif-else conditions
if 90 <= score <= 100:
    grade = "A"
elif 80 <= score < 90:
    grade = "B"
elif 70 <= score < 80:
    grade = "C"
elif 60 <= score < 70:
    grade = "D"
elif 0 <= score < 60:
    grade = "F"
else:
    grade = "Invalid score"

print(f"The letter grade for the score {score} is {grade}.")
```

The output of the code is:

```
Enter the class score (0-100): 93
The letter grade for the score 93.0 is A.
```

8. Write a code that appends the type of elements from a given list.

Input

```
x = [23, 'Python', 23.98]
```

Expected output [23, 'Python', 23.98]

```
[<class 'int'>, <class 'str'>, <class 'float'>]
```



The screenshot shows a Jupyter Notebook titled "Rushabh_Shah_ICP 1.ipynb". The code in the cell is as follows:

```
x = [23, 'Python', 23.98]

print(x)

# Create an empty list
types_list = []

# Loop through each element
for element in x:
    types_list.append(type(element))

print(types_list)
```

The output of the code is:

```
[23, 'Python', 23.98]
[<class 'int'>, <class 'str'>, <class 'float'>]
```

9. IT_companies = {'Facebook', 'Google', 'Microsoft', 'Apple', 'IBM', 'Oracle', 'Amazon'} A = {19, 22, 24, 20, 25, 26} B = {19, 22, 20, 25, 26, 24, 28, 27} age = [22, 19, 24, 25, 26, 24, 25, 24]

The screenshot shows a Jupyter Notebook titled 'Rushabh_Shah_ICP 1.ipynb'. The code defines a set of IT companies, performs various set operations (add, update, remove, union, intersection, subset, disjoint, symmetric difference), and calculates the length of the age list and its set.

```

IT_companies = {'Facebook', 'Google', 'Microsoft', 'Apple', 'IBM', 'Oracle', 'Amazon'}
A = {19, 22, 24, 20, 25, 26}
B = {19, 22, 20, 25, 26, 24, 28, 27}
age = [22, 19, 24, 25, 26, 24, 25, 24]

print("Length of IT_companies:", len(IT_companies))

IT_companies.add('Twitter')
print("After adding Twitter:", IT_companies)

IT_companies.update(['Samsung', 'Intel', 'Cisco'])
print("After adding more companies:", IT_companies)

IT_companies.remove('Oracle')
print("After removing Oracle:", IT_companies)

print("A union B:", A | B)

print("A intersection B:", A & B)

print("Is A a subset of B:", A <= B)

print("Are A and B disjoint:", A.isdisjoint(B))

print("Symmetric difference:", A ^ B)

age_set = set(age)
print("Length of age list:", len(age))
print("Length of age set:", len(age_set))

```

The output of the code is as follows:

```

Length of IT_companies: 7
After adding Twitter: {'Amazon', 'Facebook', 'IBM', 'Google', 'Twitter', 'Oracle', 'Apple', 'Microsoft'}
After adding more companies: {'Amazon', 'Twitter', 'Facebook', 'Cisco', 'IBM', 'Microsoft', 'Intel', 'Google', 'Oracle', 'Samsung', 'Apple'}
After removing Oracle: {'Amazon', 'Twitter', 'Facebook', 'Cisco', 'IBM', 'Microsoft', 'Intel', 'Google', 'Samsung', 'Apple'}
A union B: {19, 20, 22, 24, 25, 26, 27, 28}
A intersection B: {19, 20, 22, 24, 25, 26}
Is A a subset of B: True
Are A and B disjoint: False
Symmetric difference: {27, 28}
Length of age list: 8
Length of age set: 5

```