



```

In [2]: 1 #Task No:01
        2
        3 class DataType:
        4     def __init__(self, name, value) -> None:
        5         self.name = name
        6         self.value = value
        7
        8 #Qustion Part:
        9
       10 data_type1 = DataType('Integer', 1234)
       11 print(data_type1.name)
       12 print(data_type1.value)
       13 print('=====')
       14 data_type2 = DataType('String', 'Hello')
       15 print(data_type2.name)
       16 print(data_type2.value)
       17 print('=====')
       18 data_type3 = DataType('Float', 4.0)
       19 print(data_type3.name)
       20 print(data_type3.value)
       21
       22 #Output:
       23
       24 #Integer
       25 #1234
       26 #=====
       27 #String
       28 #Hello
       29 #=====
       30 #Float
       31 #4.0

```

```

Integer
1234
=====
String
Hello
=====
Float
4.0

```

```
In [2]: 1 #Task No:02
2
3 class Flower:
4     def __init__(self):
5         self.name=None
6         self.color=""
7         self.num_of_petal=0
8
9
10 #Qustion Part:
11
12 flower1 = Flower()
13 flower1.name="Rose"
14 flower1.color="Red"
15 flower1.num_of_petal=6
16 print("Name of this flower:", flower1.name)
17 print("Color of this flower:",flower1.color)
18 print("Number of petal:",flower1.num_of_petal)
19 print("=====")
20 flower2 = Flower()
21 flower2.name="Orchid"
22 flower2.color="Purple"
23 flower2.num_of_petal=4
24 print("Name of this flower:",flower2.name)
25 print("Color of this flower:",flower2.color)
26 print ("Number of petal:",flower2. num_of_petal)
27
28 #Output:
29
30 #Name of this flower: Rose
31 #Color of this flower: Red
32 #Number of petal: 6
33 #=====
34 #Name of this flower: Orchid
35 #Color of this flower: Purple
36 #Number of petal: 4
```

```
Name of this flower: Rose
Color of this flower: Red
```

Number of petal: 6

=====

Name of this flower: Orchid

Color of this flower: Purple

Number of petal: 4

```
In [1]: 1 #Task No:03
        2
        3 #Qustion Part:
        4
        5 class Wadiya():
        6     def __init__(self):
        7         self.name='Aladeen'
        8         self.designation = 'President Prime Minister Admiral General'
        9         self.num_of_wife=100
       10         self.dictator = True
       11
       12 #Output:
       13
       14 #Name of President: Aladeen
       15 #Designation: President Prime Minister Admiral General
       16 #Number of wife: 100
       17 #Is he/she a dictator: True
       18 #Part 2:
       19 #Name of President: Donald Trump
       20 #Designation: President
       21 #Number of wife: 1
       22 #Is he/she a dictator: False
       23
       24
       25
       26 w1=Wadiya()
       27 print("part1:")
       28 print(f"Name of the president: {w1.name}")
       29 print(f"Designation: {w1.designation}")
       30 print(f"Number of wife: {w1.num_of_wife}")
       31 print(f"Is he/she a dictator: {w1.dictator}")
       32
       33
       34
       35
       36 w2=Wadiya()
       37 w2.Name : "Donald Trump"
       38 w2.Designation: "President"
```

```
39 w2.num_of_wife : 1
40 w2. dictator: False
41
42
43
44 print("part2:")
45 print(f"Name of the president: {w2.name}")
46 print(f"Designation: {w2.designation}")
47 print(f"Number of wife: {w2.num_of_wife}")
48 print(f"Is he/she a dictator: {w2.dictator}")
```

part1:

Name of the president: Aladeen

Designation: President Prime Minister Admiral General

Number of wife: 100

Is he/she a dictator: True

part2:

Name of the president: Aladeen

Designation: President Prime Minister Admiral General

Number of wife: 100

Is he/she a dictator: True

```
In [2]: 1 #Task No:04
2
3 class Joker:
4     def __init__(self,a,b,c):
5         self.name=a
6         self.power=b
7         self.is_he_psycho=c
8
9
10
11
12 #Qustion Part:
13
14 j1 = Joker('Heath Ledger', 'Mind Game', False)
15 print(j1.name)
16 print(j1.power)
17 print(j1.is_he_psycho)
18 print("=====")
19 j2 = Joker('Joaquin Phoenix', 'Laughing out Loud', True)
20 print(j2.name)
21 print(j2.power)
22 print(j2.is_he_psycho)
23 print("=====")
24 if j1 == j2:
25     print('same')
26 else:
27     print('different')
28 j2.name = 'Heath Ledger'
29 if j1.name == j2.name:
30     print('same')
31 else:
32     print('different')
33
34
35 #Output:
36
37 #Heath Ledger
38 #Mind Game
```

```
39 #False
40 #=====
41 #Joaquin Phoenix
42 #Laughing out Loud
43 #True
44 #=====
45 #different
46 #same
```

Heath Ledger

Mind Game

False

=====

Joaquin Phoenix

Laughing out Loud

True

=====

different

same



```
In [1]: 1 #Task No:05
2
3 class pokemon:
4     def __init__(self,a,b,c,d,e):
5         self.pokemon1_name=a
6         self.pokemon1_power=c
7         self.pokemon2_name=b
8         self.pokemon2_power=d
9         self.damage_rate=e
10
11
12
13
14 #Qustion Part:
15
16 team_pika = pokemon('pikachu', 'charmander', 90, 60, 10)
17 print('====Team 1====')
18 print('Pokemon 1:',team_pika.pokemon1_name,team_pika.pokemon1_power)
19 print('Pokemon 2:',team_pika.pokemon2_name,team_pika.pokemon2_power)
20 pika_combined_power = (team_pika.pokemon1_power +team_pika.pokemon2_power) *
21 print('Combined Power:', pika_combined_power)
22
23
24
25 team_bulbasaur = pokemon('bulbasaur','squirtle', 80, 70, 9)
26 print('====Team 2====')
27 print('Pokemon 1:',team_bulbasaur.pokemon1_name,
28 team_bulbasaur.pokemon1_power)
29 print('Pokemon 2:',team_bulbasaur.pokemon2_name,
30 team_bulbasaur.pokemon2_power)
31 b_combined_power = (team_bulbasaur.pokemon1_power +
32 team_bulbasaur.pokemon2_power) * team_bulbasaur.damage_rate
33 print('Combined Power:', b_combined_power)
34
35
36
37 #Output:
38
```

```
39
40 #=====Team 1=====
41 #Pokemon 1: pikachu 90
42 #Pokemon 2: charmander 60
43 #Combined Power: 1500
44 #=====Team 2=====
45 #Pokemon 1: bulbasaur 80
46 #Pokemon 2: squirtle 70
47 #Combined Power: 1350
```

```
=====Team 1=====
Pokemon 1: pikachu 90
Pokemon 2: charmander 60
Combined Power: 1500
=====Team 2=====
Pokemon 1: bulbasaur 80
Pokemon 2: squirtle 70
Combined Power: 1350
```

```
In [2]: 1 #Task No:06
        2
        3 class Player:
        4     def __init__(self):
        5         self.name=None
        6         self.jersey_number=None
        7         self.position=None
        8
        9
       10
       11
       12
       13 #Qustion Part:
       14
       15
       16 player1 = Player()
       17 player1.name = "Ronaldo"
       18 player1.jersey_number = 9
       19 player1.position = "Striker"
       20 print("Name of the Player:", player1.name)
       21 print("Jersey Number of player:", player1.jersey_number)
       22 print("Position of player:", player1.position)
       23 print("=====")
       24 player2 = Player()
       25 player2.name = "Neuer"
       26 player2.jersey_number = 1
       27 player2.position = "Goal Keeper"
       28 print("Name of the player:", player2.name)
       29 print("Jersey Number of player:", player2.jersey_number)
       30 print("Position of player:", player2.position)
       31
       32
       33
       34 #Output:
       35
       36
       37 #Name of the Player: Ronaldo
       38 #Jersy Number of player: 9
```

```
39 #Position of player: Striker
40 #=====
41 #Name of the player: Neuer
42 #jersy Number of player: 1
43 #Position of player: Goal Keeper
```

```
Name of the Player: Ronaldo
Jersey Number of player: 9
Position of player: Striker
=====
Name of the player: Neuer
Jersey Number of player: 1
Position of player: Goal Keeper
```

```
In [3]: 1 #Task No:07
2
3 class country:
4     def __init__(self):
5         self.name="Bangladesh"
6         self.continent="Asia"
7         self.capital="Dhaka"
8         self.fifa_ranking=187
9
10
11
12 #Qustion Part:
13
14
15 country = country()
16 print('Name:',country.name)
17 print('Continent:',country.continent)
18 print('Capital:',country.capital)
19 print('Fifa Ranking:',country.fifa_ranking)
20 print('=====')
21 country.name = "Belgium"
22 country.continent = "Europe"
23 country.capital = "Brussels"
24 country.fifa_ranking = 1
25 print('Name:',country.name)
26 print('Continent:',country.continent)
27 print('Capital:',country.capital)
28 print('Fifa Ranking:',country.fifa_ranking)
29
30
31
32
33
34
35 #Output:
36
37 #Name: Bangladesh
38 #Continent: Asia
```

```
39 #Capital: Dhaka
40 #Fifa Ranking: 187
41 #=====
42 #Name: Belgium
43 #Continent: Europe
44 #Capital: Brussels
45 #Fifa Ranking: 1
```

```
Name: Bangladesh
Continent: Asia
Capital: Dhaka
Fifa Ranking: 187
=====
Name: Belgium
Continent: Europe
Capital: Brussels
Fifa Ranking: 1
```

```
In [4]: 1 #Task No:08
2
3 class DemonSlayer:
4     def __init__(self,a,b,c,d):
5         self.name=a
6         self.style=b
7         self.number_of_technique=c
8         self.kill=d
9
10
11 #Qustion Part:
12
13
14 tanjiro = DemonSlayer("Tanjiro", "Water Breathing", 10, 10)
15 print('Name:',tanjiro.name)
16 print('Fighting Style:',tanjiro.style)
17 print(f'Knows {tanjiro.number_of_technique} technique(s) and has killed {tanjiro.kill}')
18 print('=====')
19 zenitsu = DemonSlayer("Zenitsu", "Thunder Breathing", 1, 4)
20 print('Name:',zenitsu.name)
21 print('Fighting Style:',zenitsu.style)
22 print(f'Knows {zenitsu.number_of_technique} technique(s) and has killed {zenitsu.kill}')
23 print('=====')
24 inosuke = DemonSlayer("Inosuke", "Beast Breathing", 5, 7)
25 print('Name:',inosuke.name)
26 print('Fighting Style:',inosuke.style)
27 print(f'Knows {inosuke.number_of_technique} technique(s) and has killed {inosuke.kill}')
28 print('=====')
29 print(f'{tanjiro.name}, {zenitsu.name}, {inosuke.name} knows total {tanjiro.number_of_technique + zenitsu.number_of_technique + inosuke.number_of_technique} techniques')
30 print(f'They have killed total {tanjiro.kill + zenitsu.kill + inosuke.kill} demons')
31
32
33
34 #Output:
35
36
37 # Name: Tanjiro
38 #Fighting Style: Water Breathing
```

```

39 #Knows 10 technique(s) and has killed 10 demon(s)
40 #=====
41 #Name: Zenitsu
42 #Fighting Style: Thunder Breathing
43 #Knows 1 technique(s) and has killed 4 demon(s)
44 #=====
45
46 #Name: Inosuke
47 #Fighting Style: Beast Breathing
48 #Knows 5 technique(s) and has killed 7 demon(s)
49 #=====
50 #Tanjiro, Zenitsu, Inosuke knows total 16 techniques
51 #They have killed total 21 demons

```

Name: Tanjiro

Fighting Style: Water Breathing

Knows 10 technique(s) and has killed 10 demon(s)

=====

Name: Zenitsu

Fighting Style: Thunder Breathing

Knows 1 technique(s) and has killed 4 demon(s)

=====

Name: Inosuke

Fighting Style: Beast Breathing

Knows 5 technique(s) and has killed 7 demon(s)

=====

Tanjiro, Zenitsu, Inosuke knows total 16 techniques

They have killed total 21 demons



```
In [5]: 1 #Task No:09
2
3 class box:
4     def __init__(self,lst):
5         self.height=lst[0]
6         self.width=lst[1]
7         self.breadth=lst[2]
8         print("Creating a Box!")
9         print("Volume of the box is",self.height*self.width*self.breadth,"cul
10
11
12
13
14
15 #Qustion Part:
16
17
18 print("Box 1")
19 b1 = box([10,10,10])
20 print("=====")
21 print("Height:", b1.height)
22 print("Width:", b1.width)
23 print("Breadth:", b1.breadth)
24 print("-----")
25 print("Box 2")
26 b2 = box((30,10,10))
27 print("=====")
28 print("Height:", b2.height)
29 print("Width:", b2.width)
30 print("Breadth:", b2.breadth)
31 b2.height = 300
32 print("Updating Box 2!")
33 print("Height:", b2.height)
34 print("Width:", b2.width)
35 print("Breadth:", b2.breadth)
36 print("-----")
37 print("Box 3")
38 b3 = b2
```

```

39 print("Height:", b3.height)
40 print("Width:", b3.width)
41 print("Breadth:", b3.breadth)
42
43
44
45
46 #Output:
47
48 #Box 1
49 #Creating a Box!
50 #Volume of the box is 1000 cubic units.
51 #=====
52 #Height: 10
53 #Width: 10
54 #Breadth: 10
55 #-----
56 #Box 2
57 #Creating a Box!
58 #Volume of the box is 3000 cubic units.
59 #=====
60 #Height: 30
61 #Width: 10
62 #Breadth: 10
63 #Updating Box 2!
64 #Height: 300
65 #Width: 10
66 #Breadth: 10
67 #-----
68 #Box 3
69 #Height: 300
70 #Width: 10
71 #Breadth: 10

```

```

Box 1
Creating a Box!
Volume of the box is 1000 cubic units
=====
Height: 10
Width: 10
Breadth: 10

```

```
-----  
Box 2  
Creating a Box!  
Volume of the box is 3000 cubic units  
=====
```

Height:	30
Width:	10
Breadth:	10

```
Updating Box 2!  
Height: 300  
Width: 10  
Breadth: 10  
-----
```

Box 3
Height: 300
Width: 10
Breadth: 10

In [1]:

```

1  #Task No:10
2
3  class buttons:
4      def __init__(self, *button_data) -> None:
5          self.word, self.spaces, self.border = button_data
6          self.pera = 1 + self.spaces + len(self.word.strip()) + self.spaces +
7          print(f"{self.word} Button Specification:")
8          print(f"Button name: {self.word}")
9          print(f"Number of the border characters for the top and the bottom: ")
10         print(f"Number of spaces between the left side border and the first ")
11         print(f"Number of spaces between the right side border and the last ")
12         print(f"Characters representing the borders: {self.border}")
13         print(f"{self.border*self.pera}\n{self.border}{self.spaces*' '}{self")
14
15
16
17  #Qustion Part:
18
19
20
21  word = "CANCEL"
22  spaces = 10
23  border = 'x'
24  b1 = buttons(word, spaces, border)
25  print("=====")
26  b2 = buttons("Notify",3, '!')
27  print("=====")
28  b3 = buttons('SAVE PROGRESS', 5, '$')
29
30
31
32  #Output:
33
34  #CANCEL Button Specifications:
35  #Button name: CANCEL
36  #Number of the border characters for the top and the bottom: 28
37  #Number of spaces between the left side border and the first character of the
38  #name: 10

```

```

39 #Number of spaces between the right side border and the last character of the
40 #name: 10
41 #Characters representing the borders: x
42 #XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
43 #x CANCEL x
44 #XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
45 #=====
46
47 #Notify Button Specifications:
48 #Button name: Notify
49 #Number of the border characters for the top and the bottom: 14
50 #Number of spaces between the left side border and the first character of the
51 #name: 3
52 #Number of spaces between the right side border and the last character of the
53 #name: 3
54 #Characters representing the borders: !
55 #!!!!!!!!!!!!!!
56 #! Notify !
57 #!!!!!!!!!!!!!!
58 #=====
59 #SAVE PROGRESS Button Specifications:
60 #Button name: SAVE PROGRESS
61 #Number of the border characters for the top and the bottom: 25
62 #Number of spaces between the left side border and the first character of the
63 #name: 5
64 #Number of spaces between the right side border and the last character of the
65 #name: 5
66 #Characters representing the borders: $
67 #$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
68 # $ SAVE PROGRESS $
69 #$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

```

CANCEL Button Specification:

Button name: CANCEL

Number of the border characters for the top and the bottom: 28

Number of spaces between the left side border and the first character of the button name: 10

Number of spaces between the right side border and the last character of the button name: 10

Characters representing the borders: x

XXXXXXXXXXXXXXXXXXXXXXXXXXXXX

x CANCEL x

XXXXXXXXXXXXXXXXXXXXXXXXXXXXX

=====

Notify Button Specification:

Button name: Notify

Number of the border characters for the top and the bottom: 14

Number of spaces between the left side border and the first character of the button name: 3

Number of spaces between the right side border and the last character of the button name: 3

Characters representing the borders: !

!!!!!!!!!!!!!!

!    Notify    !

!!!!!!!!!!!!!!

=====

SAVE PROGRESS Button Specification:

Button name: SAVE PROGRESS

Number of the border characters for the top and the bottom: 25

Number of spaces between the left side border and the first character of the button name: 5

Number of spaces between the right side border and the last character of the button name: 5

Characters representing the borders: \$

\$

\$    SAVE PROGRESS    \$

\$