file1_code

February 16, 2024

0.0.1 Code for Task 1 (MCQ)

0.0.2 Rajib Dutta

0.0.3 duttarajib78@gmail.com

0.0.4 Batch DS2402

0.1 Q1

[ ]:
```
def func(a, b):
    return b if a == 0 else func(b % a, a)
print(func(30, 75))
```

```
15
```

0.2 Q2

[ ]:
```
numbers = (4, 7, 19, 2, 89, 45, 72, 22)
sorted_numbers = sorted(numbers)
even = lambda a: a % 2 == 0
even_numbers = filter(even, sorted_numbers)
print(type(even_numbers))
```

```
<class 'filter'>
```

0.3 Q3

[ ]:
```
def star_args_type_check(*args):
    print(f"The data type of the *args parameter is - {type(args)}")

star_args_type_check()
```

```
The data type of the *args parameter is - <class 'tuple'>
```

0.4 Q4

[ ]:
```
set1 = {14, 3, 55}
set2 = {82, 49, 62}
set3={99,22,17}
```

```
print(len(set1 + set2 + set3))
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[4], line 5
      2 set2 = {82, 49, 62}
      3 set3={99,22,17}
----> 5 print(len(set1 + set2 + set3))

TypeError: unsupported operand type(s) for +: 'set' and 'set'
```

0.5 Q5

[ ]:
```
def raise_exception_demo(*num_args):
    if all(isinstance(x, (int, float)) for x in num_args):
        total=0
        for anum in num_args:
            total+=anum
        return total
    else:
        raise TypeError("Type mismatch; all numbers please")

print(f"The sum of the numbers in this array = {raise_exception_demo(2, 3, 4,␣
 ↪5)}")
print(f"The sum of the numbers in this array = {raise_exception_demo(10, 12,␣
 ↪'14')}")
```

```
The sum of the numbers in this array = 14
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[5], line 11
      8         raise TypeError("Type mismatch; all numbers please")
     10 print(f"The sum of the numbers in this array = {raise_exception_demo(2,
 ↪3, 4, 5)}")
---> 11 print(f"The sum of the numbers in this array =␣
 ↪{raise_exception_demo(10, 12, '14')}")

Cell In[5], line 8, in raise_exception_demo(*num_args)
      6     return total
      7 else:
----> 8     raise TypeError("Type mismatch; all numbers please")

TypeError: Type mismatch; all numbers please
```

0.6 Q7

[ ]:
```
print(4**3 + (7 + 5)**(1 + 1))
```

208

0.7 Q10

[ ]:
```
total=0
for i in range(1, 10, 2):
    total+=i
print(f"The total = {total}")
```

The total = 25

0.8 Q16

[ ]:
```
captains = { "Enterprise": "Picard",
             "Voyager": "Janeway",
             "Defiant": "Sisko",
             }
print("Output from option a:\n")
for ship, captain in captains.items():
    print(ship, captain)

print("\n")
print("Output from option b:\n")
for ship in captains:
    print(ship, captains[ship])
```

```
Output from option a:

Enterprise Picard
Voyager Janeway
Defiant Sisko


Output from option b:

Enterprise Picard
Voyager Janeway
Defiant Sisko
```

0.9 Q17

[ ]:
```
captains = {}
captains
```

[ ]: {}

0.10 Q18

[ ]:
```
captains["Enterprise"] = "Picard"
captains["Voyager"] = "Janeway"
captains["Defiant"] = "Sisko"
captains
```

[ ]:
```
{'Enterprise': 'Picard', 'Voyager': 'Janeway', 'Defiant': 'Sisko'}
```

0.11 Q19

[ ]:
```
captains = { "Enterprise": "Picard",
             "Voyager": "Janeway",
             "Defiant": "Sisko",
             "Discovery": "unknown",
             }

for ship, captain in captains.items():
    print(f"The {ship} is captained by {captain}.")
```

```
The Enterprise is captained by Picard.
The Voyager is captained by Janeway.
The Defiant is captained by Sisko.
The Discovery is captained by unknown.
```

0.12 Q20

[ ]:
```
captains = { "Enterprise": "Picard",
             "Voyager": "Janeway",
             "Defiant": "Sisko",
             "Discovery": "unknown",
             }
del captains["Discovery"]
captains
```

[ ]:
```
{'Enterprise': 'Picard', 'Voyager': 'Janeway', 'Defiant': 'Sisko'}
```