

Name: Sowanthari R P

Date: 14.08.2024

1.Create a Java class with user defined exception handling

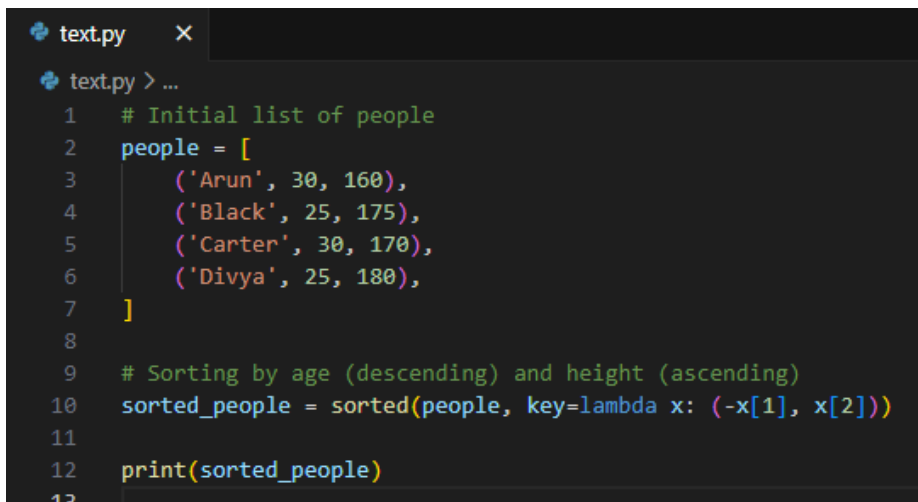
```
ExceptionExample.java
1 package PackageSample;
2
3 //Custom Exception
4 class InvalidAgeException extends Exception {
5     public InvalidAgeException(String message) {
6         super(message);
7     }
8 }
9
10 //Main Class
11 public class ExceptionExample {
12
13     public static void checkAge(int age) throws InvalidAgeException {
14         if (age < 18) {
15             throw new InvalidAgeException("Age is not valid to proceed. Must be 18 or older.");
16         } else {
17             System.out.println("Age is valid.");
18         }
19     }
20
21     public static void main(String[] args) {
22         try {
23             checkAge(15); // Change this value to test different scenarios
24         } catch (InvalidAgeException e) {
25             System.out.println("Caught the exception: " + e.getMessage());
26         }
27     }
28 }
29
30
```

Output:

```
Problems @ Javadoc Declaration Console
<terminated> ExceptionExample [Java Application] D:\Eclipse\eclipse-java-2021-06-R-win32-x86
Caught the exception: Age is not valid to proceed. Must be 18 or older.
```

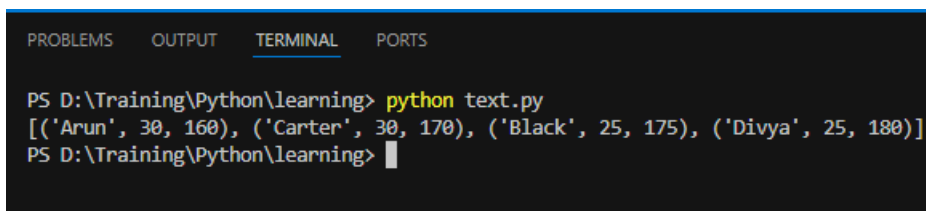
2.Modify below sorted list of user with name, age and height such that age can be descending and height as ascending using python

```
“people = [  
(&#39;Arun&#39;, 30, 160),  
(&#39;Black&#39;, 25, 175),  
(&#39;Carter&#39;, 30, 170),  
(&#39;Divya&#39;, 25, 180),  
]  
# Sort by age (ascending) and then by height (descending)  
sorted_people = sorted(people, key=lambda x: (x[1], -x[2]))  
print(sorted_people)”
```

A screenshot of a code editor window titled 'text.py'. The code defines a list 'people' with four tuples: ('Arun', 30, 160), ('Black', 25, 175), ('Carter', 30, 170), and ('Divya', 25, 180). It then sorts this list using the 'sorted' function with a lambda key: (-x[1], x[2]). The code is as follows:

```
1 # Initial list of people  
2 people = [  
3     ('Arun', 30, 160),  
4     ('Black', 25, 175),  
5     ('Carter', 30, 170),  
6     ('Divya', 25, 180),  
7 ]  
8  
9 # Sorting by age (descending) and height (ascending)  
10 sorted_people = sorted(people, key=lambda x: (-x[1], x[2]))  
11  
12 print(sorted_people)  
13
```

Output:

A screenshot of a terminal window with tabs for 'PROBLEMS', 'OUTPUT', 'TERMINAL', and 'PORTS'. The 'TERMINAL' tab is active. It shows the command 'python text.py' being executed in a PowerShell prompt at 'D:\Training\Python\learning'. The output is a list of tuples: [('Arun', 30, 160), ('Carter', 30, 170), ('Black', 25, 175), ('Divya', 25, 180)].

```
PS D:\Training\Python\learning> python text.py  
[('Arun', 30, 160), ('Carter', 30, 170), ('Black', 25, 175), ('Divya', 25, 180)]  
PS D:\Training\Python\learning>
```

3. Implement quick sort and display sorted values for [7,6,10,5,9,2,1,15,7] using python

```
text.py  X
text.py > ...
1  def quick_sort(arr):
2      if len(arr) <= 1:
3          return arr
4      pivot = arr[len(arr) // 2]
5      left = [x for x in arr if x < pivot]
6      middle = [x for x in arr if x == pivot]
7      right = [x for x in arr if x > pivot]
8      return quick_sort(left) + middle + quick_sort(right)
9
10 # Given list
11 arr = [7, 6, 10, 5, 9, 2, 1, 15, 7]
12
13 # Sorted list
14 sorted_arr = quick_sort(arr)
15 print(sorted_arr)
16
```

Output:

```
PROBLEMS  OUTPUT  TERMINAL  PORTS

PS D:\Training\Python\learning> python text.py
[1, 2, 5, 6, 7, 7, 9, 10, 15]
PS D:\Training\Python\learning> 
```