Day 14

Name : DHANAPAL

Date : 13/08/2024

1. Create a Java class with user defined exception handling
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 = [
('Arun', 30, 160),
('Black', 25, 175),
('Carter', 30, 170),
('Divya', 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)”
3. Implement quick sort and display sorted values for [7,6,10,5,9,2,1,15,7] using java or python

1. Create a Java Class with User-Defined Exception Handling

```
// Custom exception class
class AgeNotValidException extends Exception {
  public AgeNotValidException(String message) {
    super(message);
  }
}
```

```
// Main class
public class UserDefinedExceptionExample {
  public static void main(String[] args) {
    try {
      validateAge(15);  // Change the value to test
    } catch (AgeNotValidException e) {
      System.out.println("Caught the exception: " +
e.getMessage());
    }
  }

  // Method to validate age
  public static void validateAge(int age) throws
AgeNotValidException {
    if (age < 18) {
      throw new AgeNotValidException("Age is not valid for voting.
Must be 18 or above.");
    } else {
      System.out.println("Age is valid for voting.");
    }
  }
}
```

Output :

```
Caught the exception: Age is not valid for voting. Must be 18 or above.
```

2. Modify the Sorted List Using Python

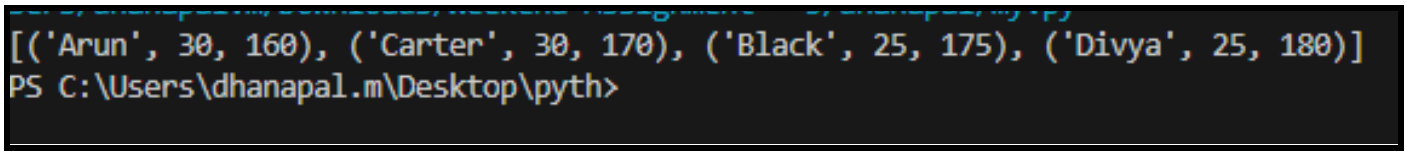
Code:

```
# Given list of people
people = [
  ('Arun', 30, 160),
  ('Black', 25, 175),
  ('Carter', 30, 170),
  ('Divya', 25, 180),
]

# Sort by age (descending) and then by height (ascending)
sorted_people = sorted(people, key=lambda x: (-x[1], x[2]))

print(sorted_people)
```

Output :



3. Implement Quick Sort in Java

Code:

```
public class QuickSortExample {

  public static void main(String[] args) {
    int[] array = {7, 6, 10, 5, 9, 2, 1, 15, 7};
```

```
    quickSort(array, 0, array.length - 1);

    System.out.print("Sorted array: ");
    for (int i : array) {
      System.out.print(i + " ");
    }
  }

  public static void quickSort(int[] array, int low, int high) {
    if (low < high) {
      int pi = partition(array, low, high);

      quickSort(array, low, pi - 1);
      quickSort(array, pi + 1, high);
    }
  }

  public static int partition(int[] array, int low, int high) {
    int pivot = array[high];
    int i = (low - 1);

    for (int j = low; j < high; j++) {
      if (array[j] <= pivot) {
        i++;

        int temp = array[i];
        array[i] = array[j];
        array[j] = temp;
      }
```

```
    }

    int temp = array[i + 1];
    array[i + 1] = array[high];
    array[high] = temp;

    return i + 1;
  }
}
```

Output :