Assignment – Module 5

1.Create a Java class with user defined exception handling?

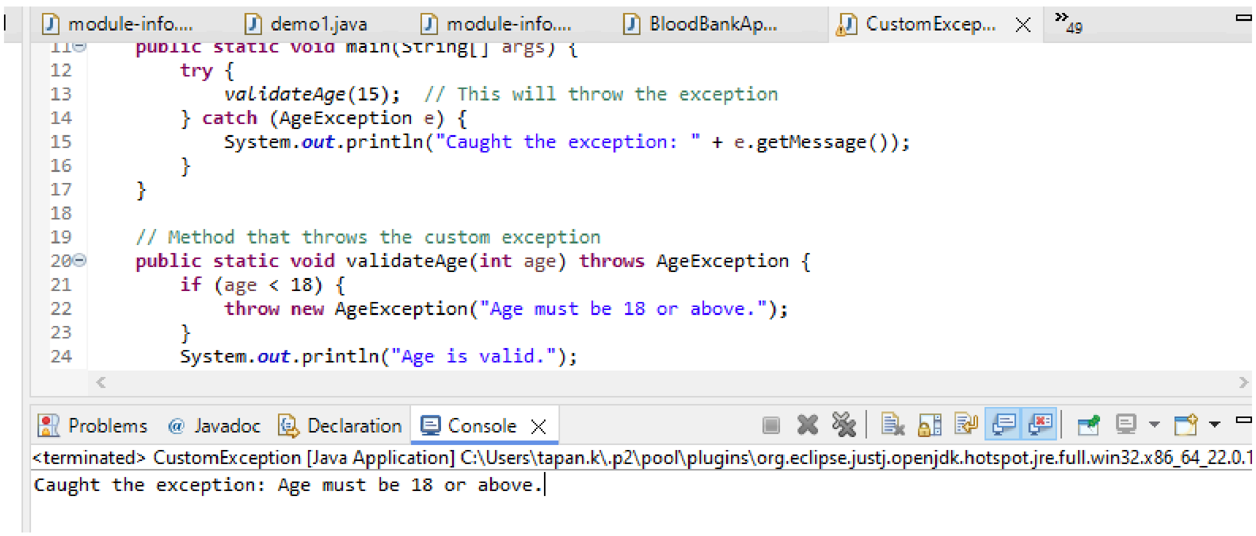
Code:-

```
class AgeException extends Exception {
    public AgeException(String message) {
        super(message);
    }
}

// Main class
public class CustomExceptionExample {
    public static void main(String[] args) {
        try {
            validateAge(15);  // This will throw the exception
        } catch (AgeException e) {
            System.out.println("Caught the exception: " + e.getMessage());
        }
    }

    // Method that throws the custom exception
    public static void validateAge(int age) throws AgeException {
        if (age < 18) {
            throw new AgeException("Age must be 18 or above.");
        }
        System.out.println("Age is valid.");
    }
}
```

Output:-



```java
    public static void main(String[] args) {
        try {
            validateAge(15);  // This will throw the exception
        } catch (AgeException e) {
            System.out.println("Caught the exception: " + e.getMessage());
        }
    }

    // Method that throws the custom exception
    public static void validateAge(int age) throws AgeException {
        if (age < 18) {
            throw new AgeException("Age must be 18 or above.");
        }
        System.out.println("Age is valid.");
```

```
<terminated> CustomException [Java Application] C:\Users\tapan.k\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_22.0.1
Caught the exception: Age must be 18 or above.
```

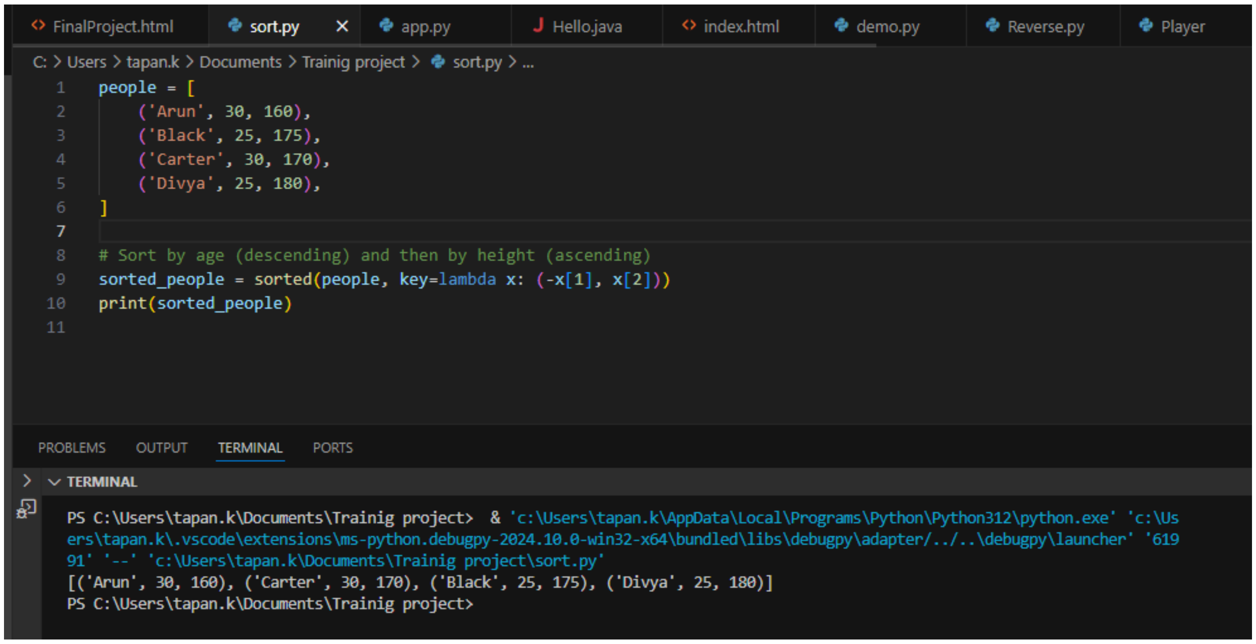
2.Modify below sorted list of user with name, age and height such that age can be descending and height as ascending using python
"people = [
('Arun', 30, 160),
('Black', 25, 175),
('Carter', 30, 170),
('Divya', 25, 180),
]
# Sort by age (ascending) and then by height (descending)
sorted_people = sorted(people, key=lambda x: (x[1], -x[2]))
print(sorted_people)"

Code:-

```python
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:-



```python
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)
```

```
PS C:\Users\tapan.k\Documents\Trainig project>  & 'c:\Users\tapan.k\AppData\Local\Programs\Python\Python312\python.exe' 'c:\Us
ers\tapan.k\.vscode\extensions\ms-python.debugpy-2024.10.0-win32-x64\bundled\libs\debugpy\adapter/../..\debugpy\launcher' '619
91' '--' 'c:\Users\tapan.k\Documents\Trainig project\sort.py'
[('Arun', 30, 160), ('Carter', 30, 170), ('Black', 25, 175), ('Divya', 25, 180)]
PS C:\Users\tapan.k\Documents\Trainig project>
```

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

Code:-

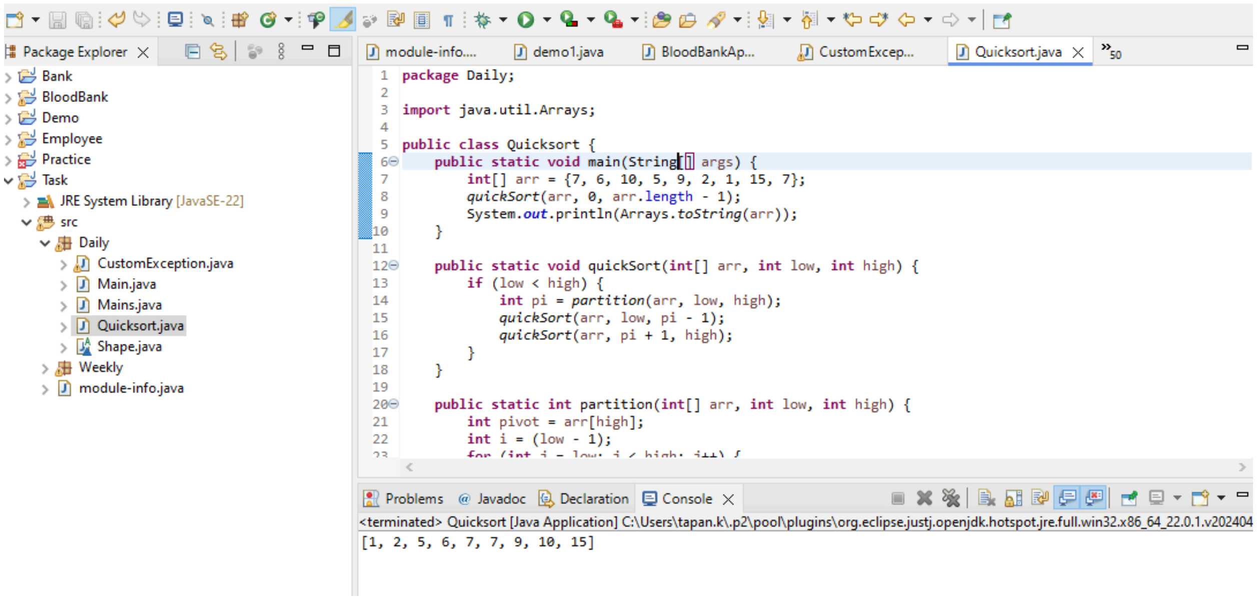
```java
import java.util.Arrays;

public class QuickSortExample {
    public static void main(String[] args) {
        int[] arr = {7, 6, 10, 5, 9, 2, 1, 15, 7};
        quickSort(arr, 0, arr.length - 1);
        System.out.println(Arrays.toString(arr));
    }

    public static void quickSort(int[] arr, int low, int high) {
        if (low < high) {
            int pi = partition(arr, low, high);
            quickSort(arr, low, pi - 1);
            quickSort(arr, pi + 1, high);
        }
    }

    public static int partition(int[] arr, int low, int high) {
        int pivot = arr[high];
        int i = (low - 1);
        for (int j = low; j < high; j++) {
            if (arr[j] <= pivot) {
                i++;
                int temp = arr[i];
                arr[i] = arr[j];
                arr[j] = temp;
            }
        }
        int temp = arr[i + 1];
        arr[i + 1] = arr[high];
        arr[high] = temp;
        return i + 1;
    }
}
```

Output:



```
package Daily;

import java.util.Arrays;

public class Quicksort {
    public static void main(String[] args) {
        int[] arr = {7, 6, 10, 5, 9, 2, 1, 15, 7};
        quickSort(arr, 0, arr.length - 1);
        System.out.println(Arrays.toString(arr));
    }

    public static void quickSort(int[] arr, int low, int high) {
        if (low < high) {
            int pi = partition(arr, low, high);
            quickSort(arr, low, pi - 1);
            quickSort(arr, pi + 1, high);
        }
    }

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

```
<terminated> Quicksort [Java Application] C:\Users\tapan.k\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_22.0.1.v202404
[1, 2, 5, 6, 7, 7, 9, 10, 15]
```