PRN: 230340325068
Set : C102

Q1 soln
Code:

```java
package com.cdac.moduleEnd;
import java.util.ArrayList;
import java.util.Collections;
import java.util.Comparator;
import java.util.Iterator;
import java.util.Scanner;
class CompareProduct implements Comparator<Product>{
    @Override
    public int compare(Product o1, Product o2) {
        // TODO Auto-generated method stub
        if (o1.getProductName().equals(o2.getProductName()) &&
o1.getQuantity()== o2.getQuantity()) {
            return 1;
        }
        return 0;
    }
}
class Product implements Comparable<Product>{
    String ProductName;
    int Quantity;

    public Product(String productName, int quantity) {
        super();
        ProductName = productName;
        Quantity = quantity;
    }
    public Product() {
        // default constructor without parameters, just to create
uninitialized object
    }

    public void setProductName(String productName) {
        ProductName = productName;
    }
    public void setQuantity(int quantity) {
        Quantity = quantity;
    }

    public String getProductName() {
        return ProductName;
    }
    public int getQuantity() {
        return Quantity;
```

```java
        }
        @Override
        public String toString() {
                return "Product [ProductName=" + ProductName + ", Quantity=" +
Quantity + "]";
        }
        @Override
        public int compareTo(Product o) {
                // TODO Auto-generated method stub
                int cmp = this.getProductName().compareTo(o.getProductName());
                if (cmp==0) {
                        cmp = this.getQuantity() - o.getQuantity();
                }
                return cmp;
        }



}
public class Inventory {
        public static void main(String[] args) {
                // TODO Auto-generated method stub

                ArrayList<Product> ProductList = new ArrayList<Product>();
                int choice = 0;
                Scanner sc = new Scanner(System.in);
                Scanner scn = new Scanner(System.in);


                do {
                        System.out.println("\n1. Add Product Details");
                        System.out.println("2. Remove Product Details");
                        System.out.println("3. Display Complete Inventory in sorted
order");
                        System.out.println("4. Exit");
                        choice = sc.nextInt();

                        switch (choice) {
                        case 1: //add products
                                System.out.println("Enter Product Details to add
product");

                                Product p1 = new Product();
                                System.out.println("Enter Product Name: ");
                                p1.setProductName(scn.nextLine());
                                System.out.println("Enter Quantity: ");
                                p1.setQuantity(sc.nextInt());

//                              ProductList.add(p1);//add single product
```

```java
                                        CompareProduct cpa = new CompareProduct();
                                        int afound=0;
                                        Iterator<Product> ait = ProductList.iterator();

                                        while(ait.hasNext()) {
                                                if(cpa.compare(p1, ait.next())==1) {
                                                        afound=1;
                                                        System.out.println("DEBUG: Product found
in add");

                                                        break;
                                                }
                                        }
                                        if (afound==1) {
                                                System.out.println("Error Adding Product: Product
Already exists");

                                        }
                                        else {
                                                ProductList.add(p1);
                                        }

                                        break;
                        case 2: //remove products
                                        System.out.println("Enter Product Details to remove
product");

                                        Product p2 = new Product();
                                        System.out.println("Enter Product Name: ");
                                        p2.setProductName(scn.nextLine());
                                        System.out.println("Enter Quantity: ");
                                        p2.setQuantity(sc.nextInt());

                                        CompareProduct cpr = new CompareProduct();
                                        Iterator<Product> rit = ProductList.iterator();
                                        int rfound=0;
//                                      ProductList.remove(p2); //to remove single object

                                        while(rit.hasNext()) {
                                                if(cpr.compare(p2, rit.next())==1) {
                                                        rfound=1;
                                                        System.out.println("DEBUG: Product found
in remove");

                                                        break;
                                                }
                                        }
                                        if(rfound==1) {
                                                rit.remove();
                                                System.out.println("Product removed
successfully");

                                        }
                                        else {
```

```java
                              System.out.println("Error Removing Product:
Product not found");
                        }

                        break;
                case 3: //display products

//                      for printing elements without sorting
//                      System.out.println("All of inventory in sorted order");
//                      for(Product oneProduct: ProductList) {
//                              System.out.println(oneProduct.toString());
//                      }

                        Collections.sort(ProductList);
                        for(Product oneProduct: ProductList) {
                                System.out.println(oneProduct.toString());
                        }

                        break;
                case 4:
                        System.out.println("Exiting App...");
                        break;
                default:
                        System.out.println("Invalid choice option");
                        break;
                }

        }while(choice !=4 );

        sc.close();
        scn.close();
    }
}
```

Runtime Screenshots:

```
1. Add Product Details
2. Remove Product Details
3. Display Complete Inventory in sorted order
4. Exit
1
Enter Product Details to add product
Enter Product Name:
Apple
Enter Quantity:
1

1. Add Product Details
2. Remove Product Details
3. Display Complete Inventory in sorted order
4. Exit
3
Product [ProductName=Apple, Quantity=1]

1. Add Product Details
2. Remove Product Details
3. Display Complete Inventory in sorted order
4. Exit
1
Enter Product Details to add product
Enter Product Name:
Mango
Enter Quantity:
5

1. Add Product Details
2. Remove Product Details
3. Display Complete Inventory in sorted order
4. Exit
1
Enter Product Details to add product
Enter Product Name:
Cherry
Enter Quantity:
3

1. Add Product Details
```

```
4. Exit
1
Enter Product Details to add product
Enter Product Name:
Cherry
Enter Quantity:
3

1. Add Product Details
2. Remove Product Details
3. Display Complete Inventory in sorted order
4. Exit
1
Enter Product Details to add product
Enter Product Name:
Cherry
Enter Quantity:
2

1. Add Product Details
2. Remove Product Details
3. Display Complete Inventory in sorted order
4. Exit
3
Product [ProductName=Apple, Quantity=1]
Product [ProductName=Cherry, Quantity=2]
Product [ProductName=Cherry, Quantity=3]
Product [ProductName=Mango, Quantity=5]

1. Add Product Details
2. Remove Product Details
3. Display Complete Inventory in sorted order
4. Exit
2
Enter Product Details to remove product
Enter Product Name:
Apple
Enter Quantity:
1
DEBUG: Product found in remove
Product removed successfully
```

Product [ProductName=Mango, Quantity=5]

1. Add Product Details
2. Remove Product Details
3. Display Complete Inventory in sorted order
4. Exit
2
Enter Product Details to remove product
Enter Product Name:
Apple
Enter Quantity:
1
DEBUG: Product found in remove
Product removed successfully

1. Add Product Details
2. Remove Product Details
3. Display Complete Inventory in sorted order
4. Exit
3
Product [ProductName=Cherry, Quantity=2]
Product [ProductName=Cherry, Quantity=3]
Product [ProductName=Mango, Quantity=5]

1. Add Product Details
2. Remove Product Details
3. Display Complete Inventory in sorted order
4. Exit
2
Enter Product Details to remove product
Enter Product Name:
Cherry
Enter Quantity:
8
Error Removing Product: Product not found

1. Add Product Details
2. Remove Product Details
3. Display Complete Inventory in sorted order
4. Exit
4
Exiting App...

Q2 soln
Code:

```java
package com.cdac.moduleEnd;
class ValueException extends Exception{
      public ValueException(String msg) {
            super(msg);
      }
}
class FitnessTracker{
      String WorkoutDuration;
      String Distance;

//      public FitnessTracker(String workoutDuration, String distance) throws
NumberFormatException, NullPointerException, ValueException
//      {
//            super();
//            this.ValidateValues(workoutDuration, distance);
//            try {
//                  Double wod = Double.parseDouble(workoutDuration);
//                  if(wod>0) {
//                        this.WorkoutDuration = String.valueOf(wod);
//                  }
//                  else {
//                        throw new ValueException("WorkoutDuration is negative
or zero");
//                  }
//
//            } catch (ValueException e) {
//                  System.out.println("Exception caught in setWorkoutDuration:
"+e.getMessage());
//            }
//            catch (NullPointerException npe) {
//                  System.out.println("Exception caught in setWorkoutDuration:
"+npe.getMessage());
//            }
//            catch (NumberFormatException nfe) {
//                  System.out.println("Exception caught in setWorkoutDuration:
"+nfe.getMessage());
//            }
////            this.setWorkoutDuration(workoutDuration);
////            WorkoutDuration = workoutDuration;
//            try {
//                  Double dist = Double.parseDouble(distance);
//                  if(dist>0) {
//                        this.Distance = String.valueOf(dist);
//                  }
//                  else {
//                        throw new ValueException("Distance is negative or
zero");
```

```java
//                      }
//              }  catch (ValueException e) {
//                      System.out.println("Exception caught in setDistance:
"+e.getMessage());
//              }catch (NullPointerException npe) {
//                      System.out.println("Exception caught in setDistance:
"+npe.getMessage());
//              }
//              catch (NumberFormatException nfe) {
//                      System.out.println("Exception caught in setDistance:
"+nfe.getMessage());
//              }
////          this.setDistance(distance);
////          Distance = distance;
//      }
      public FitnessTracker() {
              // TODO Auto-generated constructor stub
      }


      @Override
      public String toString() {
              return "FitnessTracker [WorkoutDuration=" + WorkoutDuration + ",
Distance=" + Distance + "]";
      }
      void ValidateValues(String workoutString, String distString)
      {
              try {
                      Double wod = Double.parseDouble(workoutString);
                      if(wod>0) {
                              this.WorkoutDuration = String.valueOf(wod);
                      }
                      else {
                              throw new ValueException("WorkoutDuration is negative
or zero");
                      }
              } catch (ValueException e) {
                      System.out.println("Exception caught in setWorkoutDuration:
"+e.getMessage());
              }
              catch (NullPointerException npe) {
                      System.out.println("Exception caught in setWorkoutDuration:
"+npe.getMessage());
              }
              catch (NumberFormatException nfe) {
                      System.out.println("Exception caught in setWorkoutDuration:
"+nfe.getMessage());
              }

              try {
```

```java
                Double dist = Double.parseDouble(distString);
                if(dist>0) {
                        this.Distance = String.valueOf(dist);
                }
                else {
                        throw new ValueException("Distance is negative or
zero");
                }
        } catch (ValueException e) {
                System.out.println("Exception caught in setDistance:
"+e.getMessage());
        }catch (NullPointerException npe) {
                System.out.println("Exception caught in setDistance:
"+npe.getMessage());
        }
        catch (NumberFormatException nfe) {
                System.out.println("Exception caught in setDistance:
"+nfe.getMessage());
        }


    }
}
public class FitnessTrackingTest {


    public static void main(String[] args) {
            // TODO Auto-generated method stub
            String wo, dis;
            wo="56";
            dis="100";
            FitnessTracker ft1 = new FitnessTracker();
            ft1.ValidateValues("56", "100");
            System.out.println(ft1.toString());

            FitnessTracker ft2 = new FitnessTracker();
            ft2.ValidateValues("-1", "00");
            System.out.println(ft2.toString());


    }
}
```

Runtime Screenshots:

Q3 soln
Code:

```java
package com.cdac.moduleEnd;
class Person{
    String Name;
    int Age;

    public Person(String name, int age) {
        super();
        this.Name = name;
        this.Age = age;
    }
    @Override
    public String toString() {
        return "Person [Name=" + Name + ", Age=" + Age + "]";
    }
}
class Student extends Person{
    Character Grade;
    public Student(String name, int age, Character grade) {
        super(name, age); //chaining to parent class constructor
        this.Grade = grade;
    }

    public Student(String name, int age) {
        super(name, age); //chaining to parent class constructor
    }
    @Override
    public String toString() {
        return "Student [Name=" + Name + ", Age=" + Age + ", Grade=" +
Grade + "]";
    }
}
public class PersonTest {
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Student s1 = new Student("Saman", 13);
        System.out.println(s1.toString());

        Student s2 = new Student("Danish", 15, 'A');
        System.out.println(s2.toString());

        Person p1 = new Student("Hitesh", 14);
        System.out.println(p1.toString());

        Person p2 = new Student("Rajan", 12, 'A');
        System.out.println(p2.toString());
    }
}
```

Runtime Screenshots:

```
<terminated> PersonTest [Java Application] C:\Program Files\Eclipse 4.27\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.6.v20230204-1729\jre\bin\javaw.exe  (13-May-2023, 11:29:17 am – 11:29:17 am) [pid: 2552]
Student [Name=Saman, Age=13, Grade=null]
Student [Name=Danish, Age=15, Grade=A]
Student [Name=Hitesh, Age=14, Grade=null]
Student [Name=Rajan, Age=12, Grade=A]
```