#### Name: Vaishnavi Ingole

Mail:vaishnavingole54@mail.com

# Task 1: Generics and Type Safety

Create a generic Pair class that holds two objects of different types, and write a method to return a reversed version of the pair.

```
package com.wipro.assign19;
public class Pair<T, U> {
    private T first;
    private U second;
    public Pair(T first, U second) {
        this.first = first;
        this.second = second;
    }
    public T getFirst() {
        return first;
    }
    public U getSecond() {
        return second;
    }
    public Pair<U, T> reverse() {
        return new Pair<>(second, first);
    }
    public static void main(String[] args) {
        Pair<Integer, String> intStringPair = new Pair<>(42, "Hello");
        System.out.println("Original Pair: " +
intStringPair.getFirst() + ", " + intStringPair.getSecond());
        Pair<String, Integer> reversedPair = intStringPair.reverse();
        System.out.println("Reversed Pair: " + reversedPair.getFirst()
+ ", " + reversedPair.getSecond());
}
```

```
3 public class Pair<T, U> {
4    private T first;
5    private U second;
 > A JRE System Library [JavaSE-17]

    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 

    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 
    ✓ 

    > ② Pair.java
> ② module-info.java
> ② Day5DSA
                                                                                                                                public Pair(T first, U second) {
    this.first = first;
    this.second = second;
 ∨ 🔂 Day7Dsa
         ■ JRE System Library [JavaSE-17]
      public T getFirst() {
                                                                                                                                                return first;
}
            v # com.app.pattern

    MMP.java
    MaivePatternSearching.java

                                                                                                                                                return second;
}
                                                                                                                                                 public U getSecond() {
                  ⊕ com.assign.dsa
             > # com.wipro.avl
                                                                                                                                                  public Pair<U, T> reverse() {
   return new Pair<>>([second, first)];
}
             > # com.wipro.graph
                                                                                                                          21<sup>©</sup>
22
23
             > # com.wipro.quick
             > # cpm.wipro.algo
> 1 module-info.java
 > B DSAAssignment
                                                                                                                                              public static void main(String[] args) {
    Pair<Integer, String> intStringPair = new Pair<>(42, "Hello");
     <terminated> Pair (1) [Java Application] C\Users\vaish\p2\poo\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.10.v20240120-1143\jre\bin\javaw.exe (Jun 2, 2024, 10:21:35 PM – 10:21:36 PM)
```

# Task 2: Generic Classes and Methods

Implement a generic method that swaps the positions of two elements in an array, regardless of their type, and demonstrate its usage with different object types.

```
package com.wipro.assign19;
import java.util.Arrays;

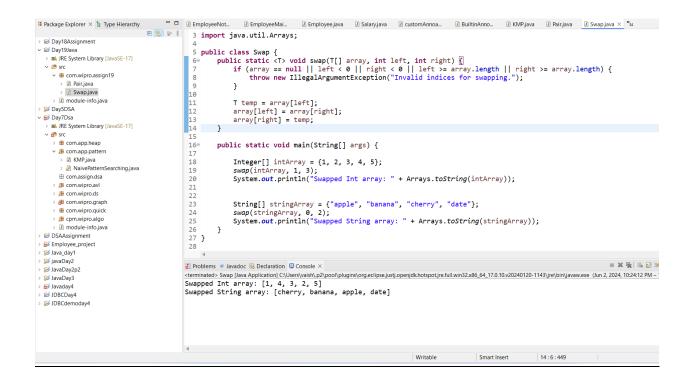
public class Swap {
    public static <T> void swap(T[] array, int left, int right) {
        if (array == null || left < 0 || right < 0 || left >=
        array.length || right >= array.length) {
            throw new IllegalArgumentException("Invalid indices for swapping.");
        }

        T temp = array[left];
        array[left] = array[right];
        array[right] = temp;
    }

    public static void main(String[] args) {
```

```
Integer[] intArray = {1, 2, 3, 4, 5};
    swap(intArray, 1, 3);
    System.out.println("Swapped Int array: " +
Arrays.toString(intArray));

    String[] stringArray = {"apple", "banana", "cherry", "date"};
    swap(stringArray, 0, 2);
    System.out.println("Swapped String array: " +
Arrays.toString(stringArray));
    }
}
```



# Task 3: Reflection API

Use reflection to inspect a class's methods, fields, and constructors, and modify the access level of a private field, setting its value during runtime

```
package com.wipro.assign19;
```

```
import java.lang.reflect.Field;
public class Reflect {
    private int PField = 42;
    public static void main(String[] args) throws
NoSuchFieldException, IllegalAccessException {
        Reflect instance = new Reflect();
        Class<?> clazz = instance.getClass();
        Field privateField = clazz.getDeclaredField("PField");
        privateField.setAccessible(true);
        int currentValue = (int) privateField.get(instance);
        System.out.println("Current value of PField: " +
currentValue);
        privateField.set(instance, 100);
        int updatedValue = (int) privateField.get(instance);
        System.out.println("Updated value of PField: " +
updatedValue);
    }
}
```

```
eclipse-workspaceWp - Day19Java/src/com/wipro/assign19/Reflect.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help

The Edit Source Refactor Navigate Search Project Run Window Help

The Edit Source Refactor Navigate Search Project Run Window Help

The Edit Source Refactor Navigate Search Project Run Window Help

The Edit Source Refactor Navigate Search Project Run Window Help

The Edit Source Refactor Navigate Search Project Run Window Help

The Edit Source Refactor Navigate Search Project Run Window Help

The Edit Source Refactor Navigate Search Project Run Window Help

The Edit Source Refactor Navigate Search Project Run Window Help

The Edit Source Refactor Navigate Search Project Run Window Help

The Edit Source Refactor Navigate Search Project Run Window Help

The Edit Source Refactor Navigate Search Project Run Window Help

The Edit Source Refactor Navigate Search Project Run Window Help

The Edit Source Refactor Navigate Search Project Run Window Help

The Edit Source Refactor Navigate Search Project Run Window Help

The Edit Source Refactor Navigate Search Project Run Window Help

The Edit Source Refactor Navigate Search Project Run Window Help

The Edit Source Refactor Navigate Search Project Run Window Help

The Edit Source Refactor Navigate Search Project Run Window Help

The Edit Source Refactor Navigate Search Project Run Window Help

The Edit Source Refactor Navigate Search Project Run Window Help

The Edit Source Refactor Project Run Window Help

The Edit Source Run Window Help

The Edit So
      > B Day18Assignment

▲ JRE System Library [JavaSE-17]
                                                                                                                                                                                                                                      5 public class Reflect {
    private int PField
                                  > 🛭 Pair.java
                                  Reflect.java
                                                                                                                                                                                                                                           8    public static void main(String[] args) throws NoSuchFieldException, IllegalAccessException {
9         Reflect instance = new Reflect();
10
                           > 1 module-info,java
   > 📂 Day5DSA

∽ 👺 Day7Dsa
                                                                                                                                                                                                                                                                                                         Class<?> clazz = instance.getClass();
              > M JRE System Library [JavaSE-17]
                                                                                                                                                                                                                                              13
14 Field privateField = clazz.getDeclaredField("PField");

∨ ∰ src
→ ∰ com.app.heap
                                                                                                                                                                                                                                                                                                            privateField.setAccessible(true);

w 

B com.app.pattern

Com.app.patte
                                  > ② KMP.java
> ② NaivePatternSearching.java
                                                                                                                                                                                                                                                                                                       int currentValue = (int) privateField.get(instance);
System.out.println("Current value of PField: " + current value of PField: 
                                   ⊕ com.assign.dsa
                           > # com.wipro.avl
                                                                                                                                                                                                                                                                                                            privateField.set(instance, 100);
                        > # com.wipro.ds
                        > # com.wipro.graph
> # com.wipro.quick
         > # cpm.wipro.algo
> 1 module-info.java
                                                                                                                                                                                                                                                                                                                         int updatedValue = (int) privateField.get(instance);
                                                                                                                                                                                                                                                                                                                         System.out.println("Updated value of PField:
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      + updatedValue);
                                                                                                                                                                                                                                                                                          }
      > # Employee project
      > 😂 Java_day1
> 😂 javaDay2
                                                                                                                                                                                                                                           Problems @ Javadoc 🚇 Declaration 📮 Console
                                                                                                                                                                                                                                           <terminated> Reflect [Java Application] C\Users\vaish\,p2\poo\\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32x86_64_17.0.10.v20240120-1143\jre\bin\javaw.exe (Jun 2, 2024, 10.27:23 PM - 10.2 Current value of PField: 42
      > 🞏 JavaDay2p2
                                                                                                                                                                                                                                           Updated value of PField: 100
        ≥ 🔐 Javaday4
           ₩ JDBCDav4
```

## Task 4: Lambda Expressions

Implement a Comparator for a Person class using a lambda expression, and sort a list of Person objects by their age..

```
package com.wipro.assign19;
import java.util.ArrayList;
import java.util.Comparator;
import java.util.List;

public class SortPerson {
    public static void main(String[] args) {
        List<Person> personList = new ArrayList<>();
        personList.add(new Person("Lily", 25));
        personList.add(new Person("Marshal", 29));
        personList.add(new Person("Robin", 22));
        personList.add(new Person("Ted", 27));

        personList.add(new Person("Ted", 27));
```

```
System.out.println("Sorted list by age:");
                                                                     for (Person person : personList) {
                                                                                         System.out.println(person.getName() +" age:"+
person.getAge());
 }
 File Edit Source Refactor Navigate Search Project Run Window Help
| The control of the
                                              ⇔ ⇔ ⊗ | E ⊗ | № 8 1 package com.wipro.assign19;
  > 🔀 Day18Assignment
                                                                                           3 import java.util.ArrayList;
4 import java.util.Comparator;
5 import java.util.List;
 > Mark JRE System Library [JavaSE-17]
     v 🕮 src
         7 public class SortPerson {
             > Pair.java
                                                                                                          public static void main(String[] args) {
               > 🛽 Person.java
                                                                                                                               list<Person> personList = new ArrayList<>();
personList.add(new Person("Lily", 25));
personList.add(new Person("Marshal", 29));
personList.add(new Person("Robin", 22));
             > 🔑 Reflect.java
              >  SortPerson.java
               >  Swap.iava
         > 1 module-info.java
 > 🎏 Day5DSA
                                                                                           13
14
15
16
17
18
19
20
21
22
                                                                                                                                personList.add(new Person("Ted", 27));
    > M JRE System Library [JavaSE-17]
         > # com.app.heap
                                                                                                                               personList.sort(Comparator.comparingInt(Person::getAge));
             > M KMP.java
              > 🕖 NaivePatternSearching.java
                                                                                                                                 System.out.println("Sorted list by age:");
                                                                                                                                 for (Person person : personList) {
    System.out.println(person.getName() +" age:"+ person.getAge());

    ⊞ com.assign.dsa

         > # com.wipro.ds
                                                                                          23
24
         > # com.wipro.graph
         > # com.wipro.quick
                                                                                            25 }
         > 🔠 cpm.wipro.algo
                                                                                           26
          > 1 module-info.java
  > 😂 DSAAssignment
                                                                                          Problems @ Javadoc Declaration Console X
 > B Employee_project
                                                                                          <a href="terminated">cterminated</a> SortPerson [Java Application] C\Users\vaish\p2\pooNplugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.10.v20240120-1143\jre\bin\javaw.exe (Jun 2, 2024)
 > 👺 Java_day1
                                                                                          Sorted list by age:
 > 📂 javaDay2
                                                                                          Robin age:22
 > 🔀 JavaDay2p2
                                                                                          Lily age:25
 > 🞏 JavaDav3
                                                                                          Ted age: 27
 ⇒ 🔒 Javaday4
                                                                                          Marshal age:29
 ⇒ IDRCDav4
 > 😂 JDBCdemoday4
```

#### **Task 5: Functional Interfaces**

Create a method that accepts functions as parameters using Predicate, Function, Consumer, and Supplier interfaces to operate on a Person object.

```
package com.wipro.assign19;
import java.util.function.Consumer;
import java.util.function.Function;
import java.util.function.Predicate;
import java.util.function.Supplier;
```

```
public class FunctionalInterf {
      public boolean testPerson(Predicate<Person> predicate, Person
person) {
             return predicate.test(person);
         }
         public <R> R applyFunction(Function<Person, R> function,
Person person) {
             return function.apply(person);
         }
         public void acceptConsumer(Consumer<Person> consumer, Person
person) {
             consumer.accept(person);
         }
         public Person getFromSupplier(Supplier<Person> supplier) {
             return supplier.get();
         }
         public static void main(String[] args) {
          FunctionalInterf operations = new
                                               FunctionalInterf();
             Person person = new Person("John Doe", 25);
             Predicate<Person> isAdult = p -> p.getAge() >= 18;
             Function<Person, String> ageBetween22And30 = p -> {
                 if (p.getAge() > 22 && p.getAge() < 30) {</pre>
                     return p.getName();
                 return null;
             };
             Consumer<Person> printPerson = p ->
System.out.println("Person: " + p.getName());
             Supplier<Person> personSupplier = () -> new
Person("Barney stinson", 28);
             if (operations.testPerson(isAdult, person)) {
```

```
System.out.println(person.getName() + " is an
adult.");
             }
             String personName =
operations.applyFunction(ageBetween22And30, person);
             if (personName != null) {
                 System.out.println(personName + " is between 22 and
30 years old.");
             operations.acceptConsumer(printPerson, person);
             Person newPerson =
operations.getFromSupplier(personSupplier);
             System.out.println("New person from supplier: " +
newPerson.getName());
             if (operations.testPerson(isAdult, newPerson)) {
                 System.out.println(newPerson.getName() + " is an
adult.");
                 personName =
operations.applyFunction(ageBetween22And30, newPerson);
                 if (personName != null) {
                     System.out.println(personName + " is between 22
and 30 years old.");
                 operations.acceptConsumer(printPerson, newPerson);
             }
         }
}
```

```
> ☑ ranjava
> ☑ Personjava
> ☑ Reflectjava
                                                        46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
                                                                                if (operations.testPerson(isAdult, person)) {
    System.out.println(person.getName() + " is an adult.");
        > 

SortPerson.java
     > 🗓 Swap.java
> 🗓 module-info.java
                                                                                String personName = operations.applyFunction(ageBetween22And30, person);
if (personName != null) {
    System.out.println(personName + " is between 22 and 30 years old.");
> 📂 Day5DSA
  operations.acceptConsumer(printPerson, person);
        > 

KMP.java

NaivePatternSearching.java
                                                                                Person newPerson = operations.getFromSupplier(personSupplier);
System.out.println[["New person from supplier: " + newPerson.getName());
     ⊕ com.assign.dsa
> # com.wipro.avl
     > # com.wipro.ds
     > # com.wipro.graph
> # com.wipro.quick
                                                                                if (operations.testPerson(isAdult, newPerson)) {
   System.out.println(newPerson.getName() + " is an adult.");
> # cpm.wipro.algo
> 1 module-info.java
> B DSAAssignment
                                                        Problems @ Javadoc Declaration Console X
                                                       > 😸 Employee_project
> 📂 javaDay2
> 😂 JavaDay2p2
> 😂 JavaDay3
> 😝 Javaday4
> 3 JDBCdemoday4
                                                                                                                                                Writable
                                                                                                                                                                                               60:82:1927
```