IntelliJ
STS (eclipse + plugins to develop Spring Boot application)
eclipse**RAD**
NetBean
Visual Studio code

IDE (Integrated development environment)

Editor
Console
IDE intellijence

****

In Java we have 3 types of valid structure:

1. Normal Class/ concreate class

```
class X{

public void fun1(){
--
}

}
```

2.abstract class

```
abstract class Y {

public void fun1(){
--
}

public abstract void fun2();

}
```

3.interface

```
public interface Z{
 void fun1();
 void fun2();
}
```

--we can define variable for above all three type of structures:

```
X x1 = new X(); // same class obj
X x1 = new XChild(); // child class obj
X x1 = null;

Y y1= new YChild();
Y y1= null;


Z z1= new ZImpl(); //implemented class object
Z z1= null;
```

Note: in order to identify the duplicate elements HashSet and LinkedHashSet class uses equals() and hashCode() method.

**All collection classes are mutually inclusive i.e we can convert any type of collection to any other collection very easily:

ex:

Demo.java:

```
package com.masai;

import java.util.ArrayList;
import java.util.HashSet;
import java.util.LinkedHashSet;
import java.util.List;
import java.util.Scanner;
import java.util.Set;

public class Demo {
	public static void main(String[] args) {

		List<Student> students = new ArrayList<Student>();


		students.add(new Student(10, "N1", 500));
		students.add(new Student(12, "N2", 520));
		students.add(new Student(13, "N3", 550));
		students.add(new Student(14, "N4", 530));
		students.add(new Student(10, "N1", 500));


		System.out.println(students.size());//5


		//converting List to Set
		Set<Student> uniqueStudents = new LinkedHashSet<>(students);

		//converting Set to List
		students = new ArrayList<Student>(uniqueStudents);

		System.out.println(students.size()); //4

	}

}
```

TreeSet:
=======

--if we try to add any elements inside the TreeSet, that element must be comparable. otherwise we get a ClassCastException.

java.lang.Comparable is an interface which have only method:

public int compareTo(Object obj);

--if we try to add any element inside the TreeSet class then that element class must implements the Comparable interface and define the sorting logic of that object by overriding compareTo method.

Note: all the Wrapper classes and String class internally implements

Comparable interface.

Note: to consider the duplicate element, HashSet and LinkedHashSet class uses equal() and hashCode() method internally, where as TreeSet class uses compareTo or compare() method.

example:

Inside the Student class:

```
	@Override
	public int compareTo(Student s2) {

//		Student s1= this;
//		
//		
//		if(s1.getMarks() > s2.getMarks())
//			return +1;
//		else
//			return -1;
//		

		return this.getMarks() > s2.getMarks() ? +1 : -1;




	}
```

Comparator(I):
==========

--this interface belongs to java.util package.
--this Comparator interface also has only one method:

public int compare(Object obj1,Object obj2);

--using this comparator we can define the sorting logic of a class objects outside that class.

--using Comparator we can define multiple sorting logic also.

StudentMarksComp.java:

```
--------------------------------
package com.masai;

import java.util.Comparator;

public class StudentMarksComp implements Comparator<Student>{

	@Override
	public int compare(Student s1, Student s2) {


		if(s1.getMarks() > s2.getMarks())
			return +1;
		else if(s1.getMarks() < s2.getMarks())
			return -1;
		else
			return s1.getName().compareTo(s2.getName());



	}



}
```

StudentRollComp.java:

```
----------------------------

package com.masai;
```

```
import java.util.Comparator;

public class StudentRollComp implements Comparator<Student>{

	@Override
	public int compare(Student s1, Student s2) {


		return s1.getRoll() > s2.getRoll() ? +1 : -1;
	}

}
```

Demo.java:

```
package com.masai;

import java.util.TreeSet;

public class Demo {



	public static void main(String[] args) {


		StudentMarksComp mcomp= new StudentMarksComp();
		StudentRollComp rcomp = new StudentRollComp();




		TreeSet<Student> ts= new TreeSet<>(mcomp);

		ts.add(new Student(10, "N1", 520));
		ts.add(new Student(12, "N2", 550));
		ts.add(new Student(6, "N3", 480));
		ts.add(new Student(5, "N4", 400));
		ts.add(new Student(14, "N5", 600));
		ts.add(new Student(15, "N6", 520));
		System.out.println(ts.size());


		for(Student student:ts) {
			System.out.println(student);
		}




	}

}
```

Student.java:

```
-----------------
package com.masai;

public class Student {

	private int roll;
	private String name;
	private int marks;

	public Student() {
		// TODO Auto-generated constructor stub
	}








	public Student(int roll, String name, int marks) {
		super();
		this.roll = roll;
		this.name = name;
		this.marks = marks;
	}
	@Override
	public String toString() {
		return "Student [roll=" + roll + ", name=" + name + ", marks=" + marks + "]";
	}


	public int getRoll() {
		return roll;
	}

	public void setRoll(int roll) {
		this.roll = roll;
	}

	public String getName() {
		return name;
	}

	public void setName(String name) {
		this.name = name;
	}

	public int getMarks() {
		return marks;
	}

	public void setMarks(int marks) {
		this.marks = marks;
	}

}
```