Collection class -

- Java collection class is used exclusively with static methods that operate on or return collections.
- It inherits Object class.
- The important points about Java Collections class are:
- Java Collection class throws a NullPointerException if the collections or class objects provided to them are null.

```
import java.util.*;
public class A5
{
   public static void main(String[] args)
   {
      ArrayList<Integer> al1 = new ArrayList<>();

      al1.add(10);
      al1.add(20);
      al1.add(30);
      al1.add(10);
      al1.add(10);
      al1.add(40);
      al1.add(-50);
      al1.add(0);
      System.out.println("ArrayList al1: "+al1);
      //ArrayList al1: [10, 20, 30, 10, 10, 40, -50, 0]

      //1. Collections.max(Collection c) -
      //It is used to get the maximum value of the given collection,
      //according to the natural ordering of its elements.

      System.out.println("Max. element in al1 is: "+Collections.max(al1));
      //Max. element in al1 is: 40

      //2. Collections.min(Collection c)
      //It is used to get the minimum value of the given collection,
      //according to the natural ordering of its elements.

      System.out.println("Max. element in al1 is: "+Collections.min(al1));
      //Max. element in al1 is: -50

      // 3. static void reverse(Collection c)
      //It is used to reverse the order of the elements in the specified list.
```

```
        Collections.reverse(al1);
        System.out.println("Reversed al1 is: "+al1);
        //Reversed al1 is: [0, -50, 40, 10, 10, 30, 20, 10]

        // 4. Collections.sort(Collection c)
        //It is used to sort the elements presents in the specified list
        //of collection inascending order.

        Collections.sort(al1);
        System.out.println("al1 Sorted in ascending order: "+al1);
        //al1 Sorted in ascending order: [-50, 0, 10, 10, 10, 20, 30, 40]

        //5. Collections.frequency(Collections c, Element e)
        //It is used to get the number of elements in the specified
        //collection equal to the specified object.

System.out.println("Frequency of 10 in al1is:"+Collections.frequency(al1,10));
        //Frequency of 10 in al1 is: 3
    }
}
```

// Use of Comperator.comparing() Method

Sort class Objects by its properties for this we use Comparator class and comparing() method in this program create getter methods for each property which are used for sorting sort by Id ,Sort by Name

```
import java.util.ArrayList;
import java.util.Iterator;
import java.util.Comparator;
import java.util.Collections;

 class Book
 {
    int bookId;
    String bookName;

    public Book(int bookId, String bookName)
    {
        this.bookId = bookId;
        this.bookName = bookName;
    }
```

```
  //getter method for Task-5
  public int getBookId()
  {
     return bookId;
  }

  //getter method for Task-4
  public String getBookName()
  {
     return bookName;
  }

  public String toString()
  {
     String msg="{id="+bookId+"::Name="+bookName+"}";
     return msg;
  }
}

class Main
{
  public static void main(String[] args)
  {
     //Task 1 adding list of book and returning the id and name
     //using toString method

     Book b1=new Book(12,"Java");
     Book b2=new Book(13,"DBMS");
     Book b3=new Book(14,"Maths");
     Book b4=new Book(15,"DS");
     System.out.println(b1);
     System.out.println(b2);
     System.out.println(b3);
     System.out.println(b4);

     //Output task-1
     /*  {id=12::Name=Java}
        {id=13::Name=DBMS}
        {id=14::Name=Maths}
        {id=15::Name=DS}       */
```

```
    //Task 2 Adding Books in Array list and printing List of books

    ArrayList<Book> l1=new ArrayList<Book>();
    l1.add(b1);
    l1.add(b2);
    l1.add(b3);
    l1.add(b4);
    System.out.println();
    System.out.println("List of Books::"+l1);
    System.out.println();

    // Output Task-2
     //List of Books::[{id=12::Name=Java}, {id=13::Name=DBMS},
{id=14::Name=Maths}, {id=15::Name=DS}]


    //Task-3 Printing ArrayList usingIterator.iterator() Method

    Iterator itr=l1.iterator();
    while(itr.hasNext())
    {
       System.out.println(itr.next());
       System.out.println();
    }
    // Task-3 Output
    /*  {id=12::Name=Java}

       {id=13::Name=DBMS}

       {id=14::Name=Maths}

       {id=15::Name=DS}    */

//Task-4 Sorting the list of books as per book name using
comperator.comparing()

    Collections.sort(l1,Comparator.comparing(Book::getBookName));

    System.out.println(">>>>>>Sorting as per Book Name<<<<<<");
    System.out.println();

    Iterator itr1=l1.iterator();
    while(itr1.hasNext())
```

```
        {
            System.out.println(itr1.next());
            System.out.println();
        }
        //OutPut Task-4
        /*>>>>>>Sorting as per Book Name<<<<<<<

            {id=13::Name=DBMS}

            {id=15::Name=DS}

            {id=12::Name=Java}

            {id=14::Name=Maths}     */

//Task-5 Sorting the list of books as per bookId using comperator.comparing()

        Collections.sort(l1,Comparator.comparing(Book::getBookId));

        System.out.println(">>>>>>Sorting as per Book ID<<<<<<<");
        System.out.println();

        Iterator itr2=l1.iterator();
        while(itr2.hasNext())
        {
            System.out.println(itr2.next());
            System.out.println();
        }

        //Output Task-5
        /*
            >>>>>>Sorting as per Book ID<<<<<<<

            {id=12::Name=Java}

            {id=13::Name=DBMS}

            {id=14::Name=Maths}

            {id=15::Name=DS}    */
    }
 }
```