1.Why collection framework in java

* Reduce the effort of coding required to write by providing useful datastructures and algorithms.
* Java collections provide high-performance and high-quality data structures and algorithms thereby increasing the speed and quality.
* Supports reusability of standard data structures and algorithms.

2.What is Collection interface?

* The Collection interface is a member of the Java Collections Framework. It is a part of java.util package.
* It is one of the root interfaces of the Collection Hierarchy. The Collection interface is not directly implemented by any class.
* However, it is implemented indirectly via its subtypes or subinterfaces like List, Queue, and Set.

3.what is package of collection framework?

java.util package

4.which is root interface of Collection Framework?

Collection interface

5.List the subinterface of Collection interface.

*List
*Queue
*Set

6.List out the classes which are implementing the List interface and Set Interface and Collection interface.

List Interface: ArrayList,LinkedList,Vector,Stack
Set Interface:HashSet,TreeSet,LinkedHashSet

7.Write a small program for adding one integer, float, double, char, string, short, boolean, byte object to list and set interface.

```
import java.util.ArrayList;
import java.util.HashSet;
import java.util.Iterator;
import java.util.List;
import java.util.ListIterator;
import java.util.Set;

public class Question_7 {

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

//Write a small program for adding one integer, float, double, char, string, short, boolean, byte object to list and set interface.

```
                        List list = new ArrayList();

                        list.add(new Integer(100));
                        list.add(new Double(123.4567890));
                        list.add(new String("Hello"));
                        list.add(new Long(23434));
                        list.add(new Boolean(true));
                        list.add(new Character('a'));
                        list.add(new Float(123.45678));
                        list.add(new Byte((byte) 128));

                        ListIterator iterator = list.listIterator();
                        System.out.println("ListInterface");
                        while (iterator.hasNext()) {
                                System.out.println(iterator.next());
                        }

                        Set set = new HashSet();

                        set.add(new Integer(100));
                        set.add(new Double(123.4567890));
                        set.add(new String("Hello"));
                        set.add(new Long(23434));
                        set.add(new Boolean(true));
                        set.add(new Character('a'));
                        set.add(new Float(123.45678));
                        set.add(new Byte((byte) 128));
                        System.out.println("-----------------------------------");
                        System.out.println("SetInterface");

                        Iterator iterator1 = set.iterator();
                        while (iterator1.hasNext()) {
                                System.out.println(iterator1.next());
                        }

        }
}
```

8.Write a program to read and print the file properties like name

of the file, size,author,dateofcreation and dateofupdation using
Properties class of collection framework.

Book.properties

```
FileName=Book.properties
FileSize=73 bytes
Author=Pavithra
DateofCreation=7/02/2023
DateofUpdation=8/02/2023
```

```
import java.io.FileReader;
import java.io.IOException;
import java.util.Properties;

public class Question_8 {

	public static void main(String[] args) throws IOException {
		// TODO Auto-generated method stub
		FileReader filereader = new FileReader(

"C:\\Users\\admin\\eclipse-workspace\\newjavaprjct\\src\\coll\\Book.properties");

		Properties p = new Properties();
		p.load(filereader);
		System.out.println(p.getProperty("FileName"));
		System.out.println(p.getProperty("FileSize"));
		System.out.println(p.getProperty("Author"));
		System.out.println(p.getProperty("DateofCreation"));
		System.out.println(p.getProperty("DateofUpdation"));
	}

}
```

9. Diffence between the List and Set

List:
1.Ordered and allow duplicate values
2.Allows the operation based on indexing

Set:
1.Unordered and does not allow duplication
2.Doesn't allow operation based on indexing
3.Doesn't maintain insertion order

10. Diffeence between ArrayList and LinkedList

ArrayList:
1.Internally uses dynamic array to the store elements
2.Manipulation is slow
3.Contiguos memory location

LinkedList:
1.Internally uses doubly linked list to store elements
2.Manipulation is faster
3.Memory loaction is not contagious

11. Difference between HashMap and HashSet

HashMap:
1.Implements Map interface
2.Allows duplicate values but keys cannot be duplicated
3.Comparatively faster than HashSet because of hashing technique
4.Uses hashing technique for adding and storing mechanism
5. Insertion method is put();

HashSet:
1.Implements Set interface
2.Doesnot allow duplicate values
3.Comparatively slower than hashmap
4.Uses HashMap object for adding and storing mechanism
5.Insertion method is add();

12. Difference between the Iterator and ListIterator

* Iterator can traverse only in forward direction whereas ListIterator traverses both in forward and backward directions.
* ListIterator can help to replace an element whereas Iterator cannot.
* Iterator helps to traverse Map, List and Set but ListIterator can only traverse List .

13.Write a program to write employee object to the file and read it.

```
import java.io.FileOutputStream;
import java.io.ObjectOutputStream;
import java.io.Serializable;

public class EmployeeFile implements Serializable {
        private int empId;
        private String empName;
        private int empAge;
        private String empMobNo;

        EmployeeFile(String name) {
```

```
}

public EmployeeFile(int empId, int empAge, String empMobNo, String empName) {
		super();
		this.empId = empId;
		this.empAge = empAge;
		this.empMobNo = empMobNo;
}

public int getEmpId() {
		return empId;
}

public void setEmpId(int empId) {
		this.empId = empId;
}

public String getEmpName() {
		return empName;
}

public void setEmpName(String empName) {
		this.empName = empName;
}

public int getAge() {
		return empAge;
}

public void setAge(int empAge) {
		this.empAge = empAge;
}

public String getMobNo() {
		return empMobNo;
}

public void setMobNo(String empMobNo) {
		this.empMobNo = empMobNo;
}

public String toString() {
		return empId + "  " + empName + "  " + empAge + "  " + empMobNo + "  ";
```

```
	}

	public static void main(String[] args) throws Exception {

			EmployeeFile employeeFile = new EmployeeFile("pavithra");

			employeeFile.setEmpId(010203);
			employeeFile.setEmpName("Pavithra");
			employeeFile.setAge(25);
			employeeFile.setMobNo("9876543123");
			employeeFile.getAge();
			employeeFile.getEmpName();
			employeeFile.getEmpId();
			employeeFile.getMobNo();
// writing object to the file
			FileOutputStream fos = new FileOutputStream("EmployeeFile.txt");
			ObjectOutputStream oos = new ObjectOutputStream(fos);
			oos.writeObject(employeeFile);
			oos.close();

	}
}
// Reading  Object from the File

   FileInputStream fis = new FileInputStream("EmployeeFile.txt");
			ObjectInputStream ois = new ObjectInputStream(fis);
			try {
				EmployeeFile employeeFile1 = (EmployeeFile) ois.readObject();
				ois.close();

				System.out.println(employeeFile1.toString());

			} catch (ClassNotFoundException e) {

				e.printStackTrace();
			} catch (IOException e) {

				e.printStackTrace();
			}
```

14.Write a program to write employee array of objects to the file and read it.

```
// Program To write employee array to the file

   import java.io.FileOutputStream;
```

```
import java.io.ObjectOutputStream;
import java.io.Serializable;
import java.util.Scanner;

public class EmployeeFile implements Serializable {
	private int empId;
	private String empName;
	private int empAge;
	private String empMobNo;

	public EmployeeFile(int empId, String empName, int empAge, String empMobNo) {
		super();
		this.empId = empId;
		this.empName = empName;
		this.empAge = empAge;
		this.empMobNo = empMobNo;
	}

	public int getEmpId() {
		return empId;
	}

	public void setEmpId(int empId) {
		this.empId = empId;
	}

	public String getEmpName() {
		return empName;
	}

	public void setEmpName(String empName) {
		this.empName = empName;
	}

	public int getAge() {
		return empAge;
	}

	public void setAge(int empAge) {
		this.empAge = empAge;
	}

	public String getMobNo() {
		return empMobNo;
```

```
	}

	public void setMobNo(String empMobNo) {
		this.empMobNo = empMobNo;
	}

	public String toString() {
		return empId + "  " + empName + "  " + empAge + "  " + empMobNo + "  ";
	}

	public static void main(String[] args) throws Exception {

		Scanner sc = new Scanner(System.in);
		EmployeeFile[] employeeFile = new EmployeeFile[3];
		System.out.println("Enter employee id,employee name,employee age
,employee number");
		for (int i = 0; i < employeeFile.length; i++) {
			employeeFile[i] = new EmployeeFile(sc.nextInt(), sc.next(),
sc.nextInt(), sc.next());

		}
		for (int i = 0; i < employeeFile.length; i++) {
			System.out.println(employeeFile[i]);

		}
		FileOutputStream fos = new FileOutputStream("EmployeeFile.txt");
		ObjectOutputStream oos = new ObjectOutputStream(fos);
		for (int i = 0; i < employeeFile.length; i++) {
			oos.writeObject(employeeFile[i]);
		}
		oos.writeObject(new EOFfile());

		oos.close();

	}
}

class EOFfile implements Serializable {

}
 // Program To read object array from the file

import java.io.FileInputStream;
import java.io.IOException;
```

```
import java.io.ObjectInputStream;

public class EmpobjRead {

	public static void main(String[] args) throws IOException, ClassNotFoundException {
		// TODO Auto-generated method stub

		FileInputStream fis = new FileInputStream("EmployeeFile.txt");
		ObjectInputStream ois = new ObjectInputStream(fis);
		try {

			Object obj=null;
			while((obj=ois.readObject()) instanceof EOFfile==false)
			{
				EmployeeFile pr=(EmployeeFile)obj;
			System.out.println(pr);
			}

			ois.close();


		} catch (IOException e) {

			e.printStackTrace();
		}

	}

}
```