JAVA
ASSIGNMENT- 5 (24)

R. DEEPAK
192325027

Collection in java

A collection represents a single unit of objects

1) List ii) Queue iii) Set

ArrayList	priority queue	Hashset
LinkedList	Deque	LinkedHashset
vector	ArrayQueue	Sorted Set
Stack		Tree set

i) ArrayList

```
import java.util.*;
class Main {
public static void main(String[]args){
ArrayList<String>list = new ArrayList<>();
List.add("Ravi");
List.add("Vijay");
List.add("Ajay");
Iterator itr = list.iterator();
while(itr.hasNext()){
        System.out.println(itr.next());
}}}
```

Output : Ravi
Vijay
Ajay

Linked List

```
import java.util.*;
class Main{
public static void main(String[] args){
LinkedList<String>S = new LinkedList<>()
S.add("Ajay");
S.add("Ravi");
S.add("Vijay");
```

```
Iterator < String > itr = s. iterator ();
while (itr. hasNext ()){
    System.out. printer (itr. Next());
}}}
```

iii) Vector

```
import java. util.* ;
class Main{
  Public Static void Main( String [] args){
  vector [String> v = new vector <>();
  v. add ("1");
  v. add ("2");
  v. add ("3");
  Iterator & string > itr = v. iterator ();
  while (itr. has Next()){
      System. out. printer (itr. next());
}}}
```

Output : 1
2
3

Stack

```
import java. util .* ;
class Main{
  public static void Main(String [] args){
  Stack < string > s = new stack <>();
  S. push ("1");
  S. push ("2");
  S. push ("3");
  S. push ("4");
  Stack. pop()
  Iterator< string > itr = S. iterator ();
  while (itr. has next ()){
      System. out. printer (itr. next ());
}}}
```

Output : 1
2
3

v) priority Queue :

```
import java.util.*;
class Main {
    public static void main(String[] args) {
    priority Queue<String> queue =
                new priority Queue<>();

    queue.add("1");
    queue.add("2");
    queue.add("3");
    System.out.println("need" + queue.element());
    System.out.println("need" + queue.peek());
    System.out.println("Iterating elements:");
    Iterator i = queue.iterator();
    while (i.hasNext()) {
        System.out.println(i.next());
}}}
```

output :
need : Amit
need : Amit
Iterating elements :
1
2
3.

vi) Deque :

```
import java.util.*;
class Main {
    public static void main(String[] arg
  Deque<String> d = new ArrayDeque<St
d.add("1");
d.add("2");
d.add("3");
```

```
for (String str deque) {
    System.out.println (str);
}
}
```

Output : 1
2
3

vii) treeset :

```
import java.util.*
class Main {
    public static void main (String[] args) {
        Treeset <String> s = new Treeset <> ();
        s.add ("1");
        s.add ("2");
        s.add ("3");
        Iterator <string> i = s.iterator()
        while (i.hasnext ()) {
            System.out.println (i.next());
    }
}
```

output : 1
2
3

ii) hash set :-

```
import java.util.*;
class main {
    public static void main (String [] args) {
    Hashset <string> set = new Hashset <> ();
    Set.add ("Ravi");
    Set.add ("Raj");
    set.add ("gopi");
    Iterator <String> i = set.iterator ();
    while (i.hasnext()) {
        System.out.println (i.next());
}
}
```

Linked Hashset

```
import java.util.*;
class Main {
    public static void main (String[] args) {
    LinkedHashSet<String> set = new LinkedHashSet
                                <>();
    Set.add("1");
    Set.add("2");
    Set.add("3");
    Set.add("1");
    Iterator<String> i = set.iterator();
    while(i.hasNext()) {
        System.out.println(i.next());
    }
}
}
```

x) **Sorted Set:**

```
import java.util.*;
class Main {
    public static void main(String[] args)
    {
    SortedSet<String> s = new TreeSet<String>();
      s.add("One");
      s.add("two");
      s.add("three");
      System.out.println("set after inserting" + s
      s.remove("two");
      System.out.println("removed" + s);
    }
}
```