

Q.1) Write a program to display date and time in java ?

Ans:-

```
public class DateAndtimeApi {  
  
    public static void main(String[] args) {  
        LocalDate date= LocalDate.now();  
        LocalTime time= LocalTime.now();  
  
        System.out.println(date);  
        System.out.println(time);  
    }  
}
```

Q.2) Write a program to convert a date into String in the format of DD/mm/YYYY?

Ans:-

```
public class DateAndtimeApi {  
  
    public static void main(String[] args) {  
        LocalDate date= LocalDate.now();  
        LocalTime time= LocalTime.now();  
  
        System.out.println(date);  
        System.out.println(time);  
  
        int DD= date.getDayOfMonth();  
        int mm= date.getMonthValue();  
        int yy= date.getYear();  
  
        System.out.println(DD + "/" + mm + "/" + yy);  
    }  
}
```

Q.3) What is the difference between collection and stream ? explain with an example.

Ans:- Collection :- It is a group of individual objects as a single entity.

It is a data structure that store elements.

```
public static void main(String[] args) {  
  
    ArrayList list1 = new ArrayList();  
    list1.add(100);  
    list1.add(400);  
    list1.add(700);  
    System.out.println(list1);  
}
```

Stream :- It is the process of group of objects from the collection.

It is not a data structure , it conveys elements from a source.

```
public static void main(String[] args) {  
  
    List<Integer> list1 = Arrays.asList(5, 8, 2, 9, 7);  
    System.out.println(list1);  
  
    Stream<Integer> stream1 = list1.stream();  
    stream1.forEach(n -> System.out.println(n));  
}
```

Q.4) What is enum in java ? Explain with example.

Ans:- It is used to define a group of named constant. For example –

```
package APIsAndAnnotation;  
// it is a functional interface .  
  
@FunctionalInterface  
interface demo {  
    void flying();  
    // void flying2();  
}  
  
@Deprecated // this means , it is not possible to create objects of this class  
class plane {  
    public void Fly() {  
        System.out.println("plane is flying");  
    }  
}  
  
class cargoPlane extends plane {  
    // this is a overiden method  
    @Override  
    public void Fly() {  
        System.out.println("cargoplane is flying");  
    }  
}
```

```
public class Annotation1 {  
    public static void main(String[] args) {  
        plane p = new cargoPlane();  
        p.Fly();  
    }  
}
```

Q.5) What are in built annotations ?

Ans:- java provides several built-in annotations that can be used to provide additional information about code elements. Here are some of the most commonly used built-in annotations in Java:

Such as , @Override , @Retention, @Target etc.