

Program to display current date and time in java?

```
// Program to display current date and time in Java

import java.time.LocalDateTime;

import java.time.format.DateTimeFormatter;

public class CurrentDateTime {

    public static void main(String[] args) {

        LocalDateTime currentDateTime = LocalDateTime.now();

        System.out.println("Current Date and Time: " + currentDateTime);

    }

}
```

Write a program to convert a date to a string in the format "MM/dd/yyyy"

```
import java.time.LocalDate;

import java.time.format.DateTimeFormatter;

public class DateToString {

    public static void main(String[] args) {

        LocalDate date = LocalDate.now();

        DateTimeFormatter formatter = DateTimeFormatter.ofPattern("MM/dd/yyyy");

        String formattedDate = date.format(formatter);

        System.out.println("Formatted Date: " + formattedDate)

    } }
```

What is the difference between collections and streams? Explain with an Example

<u>Collection</u>	<u>Stream</u>
Collections: Store data in memory and allow direct access to elements.	Allow data to be processed in a pipeline without storing them in memory.
Example: List, Set, Map.	Example: Filtering and printing elements using streams.
<u>// Collections Example</u> import java.util.ArrayList; import java.util.List; public class CollectionsExample { public static void main(String[] args) { List<String> names = new ArrayList<>(); names.add("Alice"); names.add("Bob"); names.add("Charlie"); for (String name : names) { System.out.println(name); } } } }	<u>// Streams Example</u> import java.util.stream.Stream; public class StreamsExample { public static void main(String[] args) { Stream<String> namesStream = Stream.of("Alice", "Bob", "Charlie"); namesStream.forEach(System.out::println); } }

What is enums in java? explain with an example

Enums are special Java types used to define collections of constants.

Example: Days of the week, where each day is a constant.

```
enum Day{
    SUNDAY, MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, SATURDAY
}

public class EnumExample {
    public static void main(String[] args) {
        Day today = Day.WEDNESDAY;
        switch (today) {
            case MONDAY:
                System.out.println("Today is Monday");
                break;
            case WEDNESDAY:
                System.out.println("Today is Wednesday");
                Break;
            case FRIDAY:
                System.out.println("Today is Friday");
                break;
            Default:
                System.out.println("Today is some other day");
                break;
        }
    }
}
```

```
    }  
    }  
}
```

What are in built annotations in java?

In-built annotations in Java

In-built annotations in Java include:

1. `@Override` - Indicates that a method overrides a method in a superclass.
2. `@Deprecated` - Marks a method or class as deprecated, indicating it should not be used.
3. `@SuppressWarnings` - Suppresses compiler warnings for the annotated element.

Example:

```
class Parent {  
  
    @Deprecated  
  
    void display() {  
  
        System.out.println("Parent display method");  
  
    }  
  
}  
  
class Child extends Parent {  
  
    @Override  
  
    void display() {  
  
        System.out.println("Child display method");  
  
    }  
  
}
```

```
@SuppressWarnings("deprecation")

public static void main(String[] args) {

    Child obj = new Child();

    obj.display();

}

}
```