**APIs and Annotation-Assignment**

Question 1: -**Program to display current date and time in java.**

Program: -

**package** PWskill\_API;

**import** java.sql.Date;

**import** java.time.LocalDate;

**public** **class** current\_date {

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

java.util.Date dt=**new** java.util.Date();

System.***out***.println(dt);

// LocalDate ld=LocalDate.now();

// System.out.println(ld.getDayOfMonth());

}

}

Question 2: -**Write a Program to convert a date to a string in format “MM/DD//YY”.**

Program: -

**package** PWskill\_API;

**import** java.sql.Date;

**import** java.text.SimpleDateFormat;

**import** java.time.LocalDate;

**import** oracle.sql.DATE;

**public** **class** date\_2\_String {

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

// LocalDate ld = LocalDate.now();

// System.out.println(ld.getDayOfMonth() + "" + ld.getMonth());

java.util.Date dt = **new** java.util.Date();

SimpleDateFormat sdf = **new** SimpleDateFormat("MM/DD/YY");

String str = sdf.format(dt);

System.***out***.println(str);

}

}

Question 3: -**What are the difference between collections streams?Explain with an example.**

Ans: -

| **Sr. No.** | **Key** | **Collections** | **Streams** |
| --- | --- | --- | --- |
| 1 | Basic | It is used for storing and manipulating group of data | Stream API is only used for processing group of data |
| 2 | Package | All the classes and interfaces of this API is in the Java.util package | All the classes and interfaces of this API is in the java.util.stream package |
| 3 | Eager/Lazy | All the elements in the collections are computed in the beginning. | In streams, intermediate operations are lazy. |
| 4. | Data Modification | In collections, we can remove or add elements. | We can’t modify streams. |
| 5 | External /Internal iterator | Collections perform iteration over the collection. | Stream perform iteration internally. |

**Example: -**

**package** PWskill\_API;

**import** java.util.ArrayList;

**import** java.util.Collections;

**import** java.util.List;

**public** **class** difference {

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

List<String> list=**new** ArrayList<>();

list.add("vikash");

list.add("Vivek");

list.add("Smile");

//Using collections class......

Collections.*sort*(list);

**for**(String name:list) {

System.***out***.println(name);

}

System.***out***.println(">>>>>>>>>>>>>>>");

//using Streams ..........

//list.stream().sorted().forEach(n->System.out.println(n));

list.stream().sorted().forEach(System.***out***::println);

}

}

Question 4: -**What are the enums in java?Explain with an example.**

Ans: -An enum type is a special data type that enables for a variable to be a set of predefined constants. The variable must be equal to one of the values that have been predefined for it. Common examples include compass directions (values of NORTH, SOUTH, EAST, and WEST) and the days of the week.

Some major point about enums are:

* Enum improves type safety
* Enum can be easily used in switch
* Enum can be traversed
* Enum can have fields, constructors and methods
* Enum may implement many interfaces but cannot extend any class because it internally extends Enum class

**Example: -**

**package PWskill\_API;**

**enum day {**

***SUNDAY*, *MONDAY*, *TUESDAY*, *WEDNESDAY*, *THURSDAY*, *FRIDAY*, *SATURDAY*;**

**}**

**public class enums\_example {**

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

**day sunDay = day.*SUNDAY*;**

**switch (sunDay) {**

**case *SUNDAY*:**

**System.*out*.println("SUNDAY");**

**break;**

**case *MONDAY*:**

**System.*out*.println("monday");**

**break;**

**default:**

**System.*out*.println("other day");**

**break;**

**}**

**}}**

Question 5: -**What are in built annotations in java?.**

Ans: -Annotations in [Java](https://www.simplilearn.com/tutorials/java-tutorial/what-is-java) provide additional information to the compiler and JVM. An annotation is a tag representing metadata about classes, interfaces, variables, methods, or fields. Annotations do not impact the execution of the code that they annotate.

Annotations in Java are used to provide additional information, so it is an alternative option for XML and Java marker interfaces.

Built-In Java Annotations used in Java code

* **@Override**
* **@SuppressWarnings**
* **@Deprecated**

## Built-In Java Annotations used in other annotations

* **@Target**
* **@Retention**
* **@Inherited**
* **@Documented**

**Example: -**

**package PWskill\_API;**

**import java.lang.annotation.Annotation;**

**import java.lang.annotation.ElementType;**

**import java.lang.annotation.Retention;**

**import java.lang.annotation.RetentionPolicy;**

**import java.lang.annotation.Target;**

**@Target(ElementType.*TYPE*)**

**@Retention(RetentionPolicy.*RUNTIME*)**

**@interface player{**

**String country() default "Indian";**

**int age() default 22;**

**}**

**@player**

**class virat{**

**private int inning;**

**private int run;**

**public int getInning() {**

**return inning;**

**}**

**public void setInning(int inning) {**

**this.inning = inning;**

**}**

**public int getRun() {**

**return run;**

**}**

**public void setRun(int run) {**

**this.run = run;**

**}**

**}**

**public class custom\_annotations {**

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

**virat v=new virat();**

**v.setInning(200);**

**v.setRun(30000);**

**System.*out*.println(v.getInning());**

**System.*out*.println(v.getRun());**

**Class c=v.getClass();**

**//System.out.println(c);**

**Annotation annotation=c.getAnnotation(player.class);**

**player values=(player)annotation;**

**String country=values.country();**

**int age=values.age();**

**System.*out*.println("Country is:"+country+"\n age is :"+age);**

**}**

**}**