

## Java Lab 15

1. Write a Java program to get the dates 10 days before and after today.

```
import java.time.LocalDate;
```

```
public class Q1_dateAPI {
```

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

```
        LocalDate ID = LocalDate.now();
```

```
        System.out.println("Today's date : "+ID);
```

```
        System.out.println("-----");
```

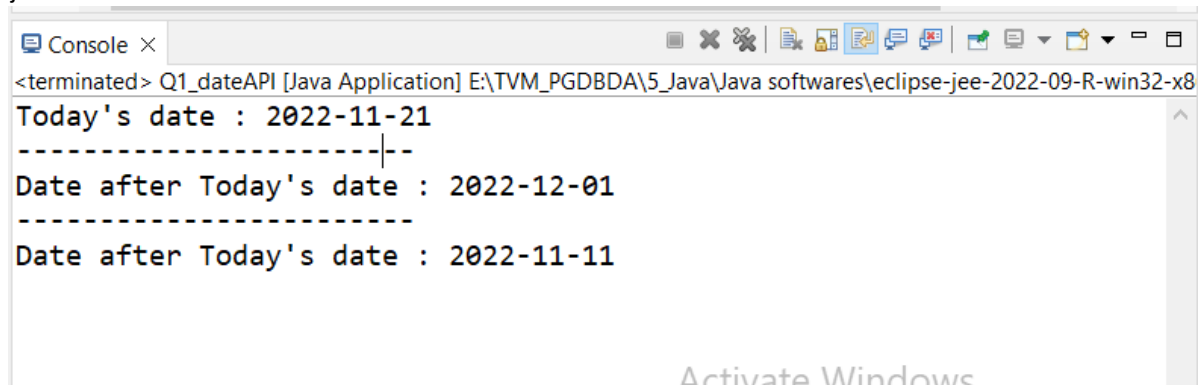
```
        System.out.println("Date after Today's date : "+ID.plusDays(10));
```

```
        System.out.println("-----");
```

```
        System.out.println("Date after Today's date : "+ID.minusDays(10));
```

```
    }
```

```
}
```



The screenshot shows a console window titled "Console" with the following output:

```
<terminated> Q1_dateAPI [Java Application] E:\TVM_PGDBDA\5_Java\Java softwares\eclipse-jee-2022-09-R-win32-x86
Today's date : 2022-11-21
-----
Date after Today's date : 2022-12-01
-----
Date after Today's date : 2022-11-11
```

An "Activate Windows" watermark is visible in the bottom right corner of the console window.

2. Write a program to create a simple Server in your machine. The server will send a message "Hello Everyone" whenever a client connects to it.

```
import java.net.*;
```

```
import java.io.*;
```

```
class MyServer{
```

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

```
ServerSocket ss = new ServerSocket(3333);
```

```
Socket s = ss.accept();
```

```
DataInputStream din=new DataInputStream(s.getInputStream());
```

```
DataOutputStream dout=new DataOutputStream(s.getOutputStream());
```

```
BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
```

```
String str="",str2="";
```

```
System.out.println("client connected");
```

```
while(!str.equals("stop")){
```

```
    str=din.readUTF();
```

```
    System.out.println("client says: "+str);
```

```
    str2=br.readLine();
```

```
    dout.writeUTF(str2);
```

```
    dout.flush();
```

```
}
```

```
din.close();
```

```
s.close();
```

```
ss.close();
```

```
}}
```

```
import java.net.*;
```

```
import java.io.*;
```

```
class MyClient{
```

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

```
Socket s=new Socket("localhost",3333);

DataInputStream din=new DataInputStream(s.getInputStream());
DataOutputStream dout=new DataOutputStream(s.getOutputStream());

BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

String str="",str2="";

System.out.println("connected to server");

while(!str.equals("stop")){
    str=br.readLine();
    dout.writeUTF(str);
    dout.flush();
    str2=din.readUTF();
    System.out.println("Server says: "+str2);
}

    dout.close();
    s.close();

}}
```

```
C:\Windows\System32\cmd.exe - JAVA MyClient
use --help for a list of possible options

E:\TVM_PGDBDA\5_Java\Java_class\Day_17_21st_Nov>JAVAC MyClient.java

E:\TVM_PGDBDA\5_Java\Java_class\Day_17_21st_Nov>JAVA MyClient
Exception in thread "main" java.net.ConnectException: Connection refused: connect
    at java.base/sun.nio.ch.Net.connect0(Native Method)
    at java.base/sun.nio.ch.Net.connect(Net.java:579)
    at java.base/sun.nio.ch.Net.connect(Net.java:568)
    at java.base/sun.nio.ch.NioSocketImpl.connect(NioSocketImpl.java:576)
    at java.base/java.net.SocksSocketImpl.connect(SocksSocketImpl.java:327)
    at java.base/java.net.Socket.connect(Socket.java:666)
    at java.base/java.net.Socket.connect(Socket.java:600)
    at java.base/java.net.Socket.<init>(Socket.java:509)
    at java.base/java.net.Socket.<init>(Socket.java:289)
    at MyClient.main(MyClient.java:8)

E:\TVM_PGDBDA\5_Java\Java_class\Day_17_21st_Nov>JAVA MyClient
connected to server
Hi my self rajan, client.
Server says: Hi Rajan, welcome to CDAC.

C:\Windows\System32\cmd.exe - Java MyServer
Microsoft Windows [Version 10.0.19044.2251]
(c) Microsoft Corporation. All rights reserved.

E:\TVM_PGDBDA\5_Java\Java_class\Day_17_21st_Nov>Javac MyServer.java

E:\TVM_PGDBDA\5_Java\Java_class\Day_17_21st_Nov>Java MyServer
client connected
client says: Hi my self rajan, client.
Hi Rajan, welcome to CDAC.
```

3. Create Product POJO with data members id, name, category and price. Create a list of Products and do the following operations using Stream.
- a) Get a list of products which belongs to category "Books" with price > 100.
  - b) Get the total no: of products.

>>

```
package com.pojoAssi_15.pojo;
```

```
public class product {
    private int id;
    private String name;
    private String catagory;
    private int price;
    public product(int id, String name, String catagory, int price) {

        this.id = id;
        this.name = name;
        this.catagory = catagory;
        this.price = price;
    }

    public product() {

    }

    public int getId() {
        return id;
    }
    public void setId(int id) {
        this.id = id;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public String getCatagory() {
        return catagory;
    }
    public void setCatagory(String catagory) {
        this.catagory = catagory;
    }
    public int getPrice() {
        return price;
    }
}
```

```

    }
    public void setPrice(int price) {
        this.price = price;
    }
}

```

```

import java.util.ArrayList;
import java.util.List;

```

```

import com.pojoAssi_15.pojo.product;

```

```

public class Q3main {

    public static void main(String[] args) {
        System.out.println("Books whose price is greater than Rs.100");

        List<product> plist = new ArrayList<product>();

        plist.add(new product(1, "Wings of fire", "autobiography", 500));

        plist.add(new product(2, "Malgudi", "fiction", 2500));

        plist.add(new product(3, "Ramayana", "Devotional", 500));

        plist.add(new product(4, "Mahabharat", "Devotional", 3800));

        plist.add(new product(5, "The Great Indian Noval", "History", 1550));

        plist.add(new product(6, "Half girlfriend ", "Romance", 1500));

        plist.stream().filter(p -> p.getPrice()>100) //filtering price
        .map(pm -> pm.getPrice()) //fetching price.
        .forEach(System.out:: println);

        System.out.println("Total number of element are :-"+plist.stream().count());

    }
}

```

```
Console ×
<terminated> Q3main (1) [Java Application] E:\TVM_PGDBDA\5_Java\Java softwares\ecli
Books whose price is greater than Rs.100
500
2500
500
3800
1550
1500
Total number of element are :-6
```