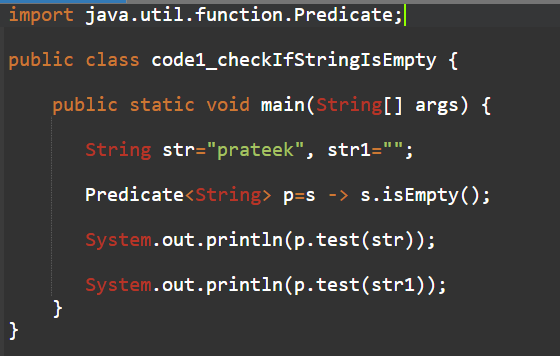
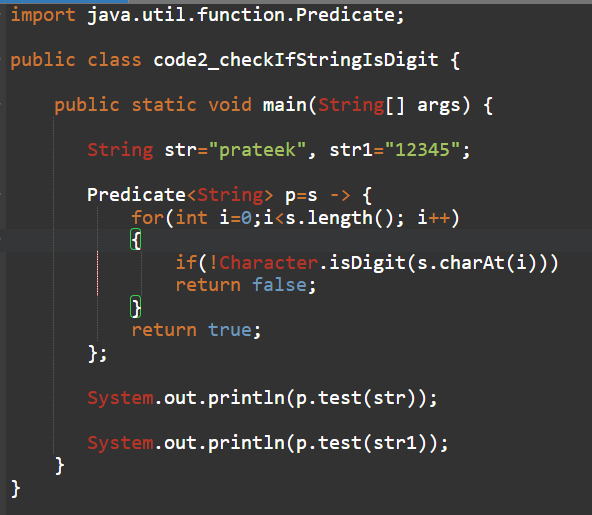
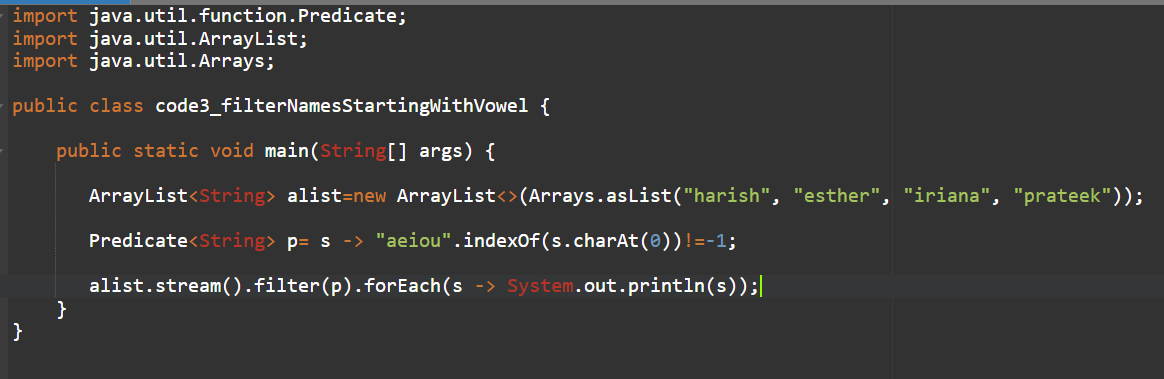
1. **Check if string is empty**

****

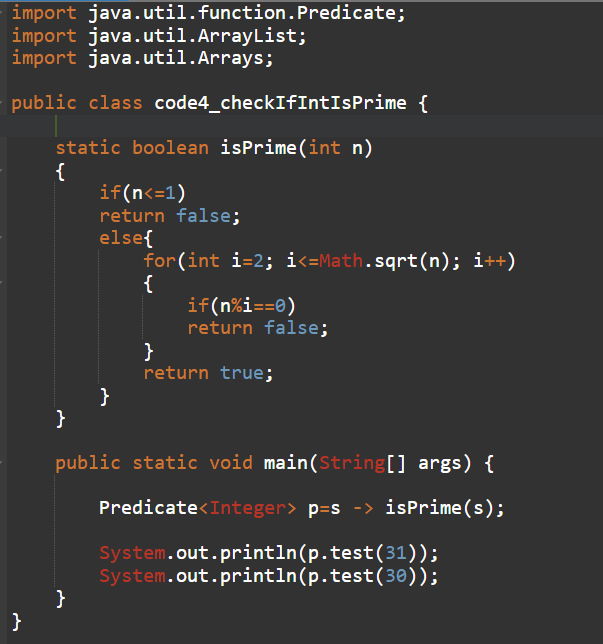
1. **Check if string is digits**

****

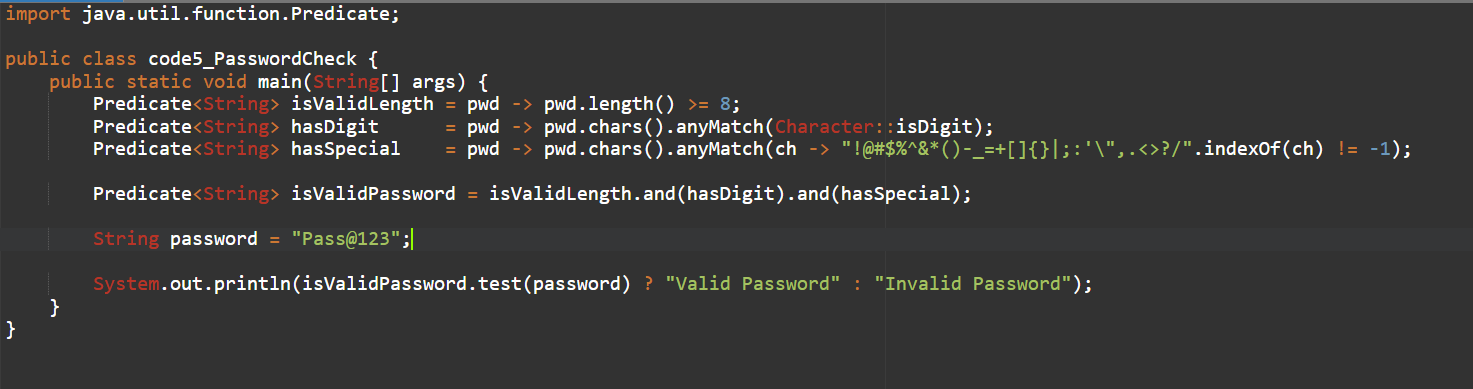
1. **Filter out names that start with a vowel from a list.**

****

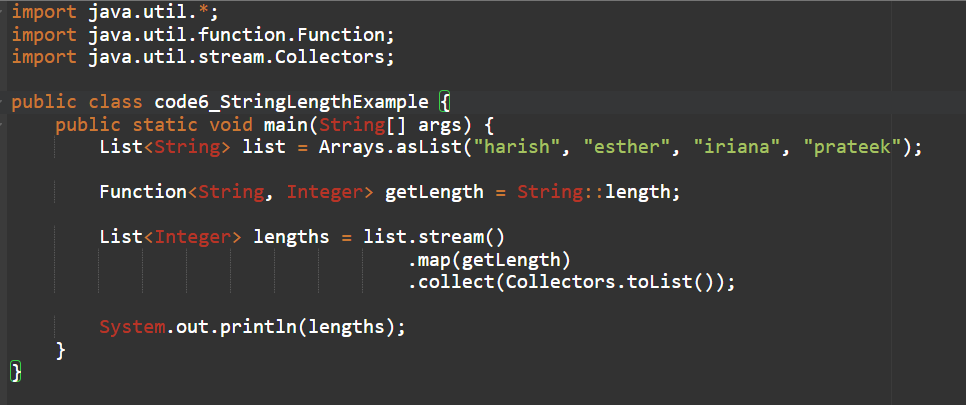
1. **Validate if a number is prime.**

****

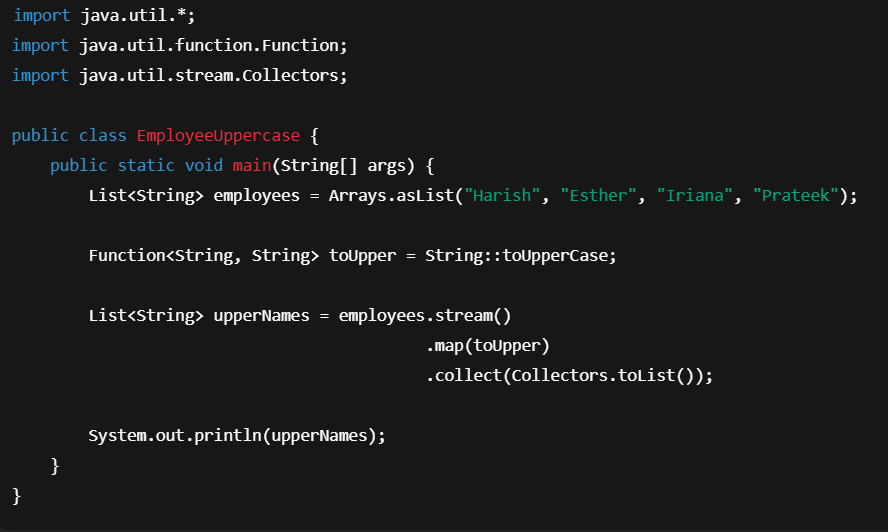
1. **Check if a password meets complexity rules (length, digit, special char).**

****

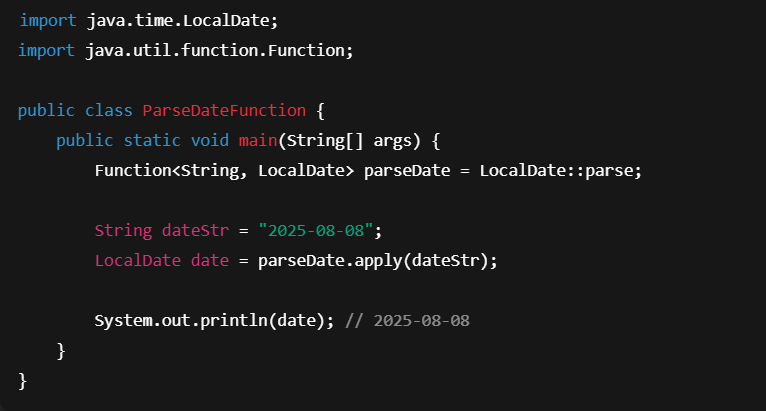
1. **Convert a list of strings to their lengths.**

****

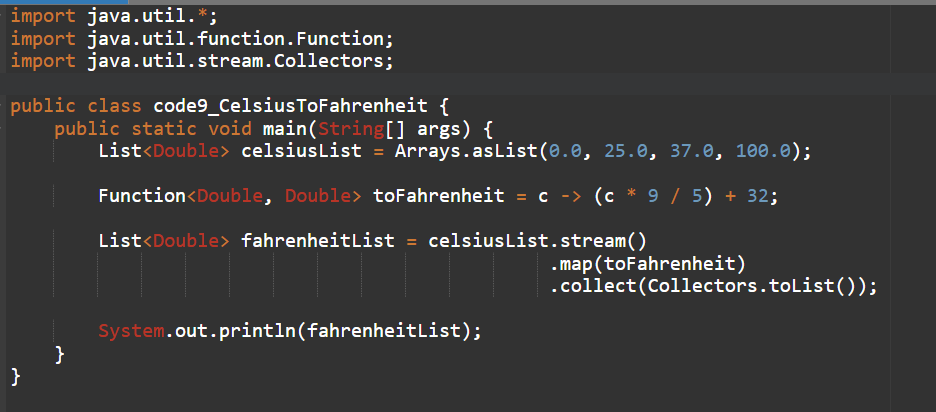
1. **Map employee names to their uppercase versions.**

****

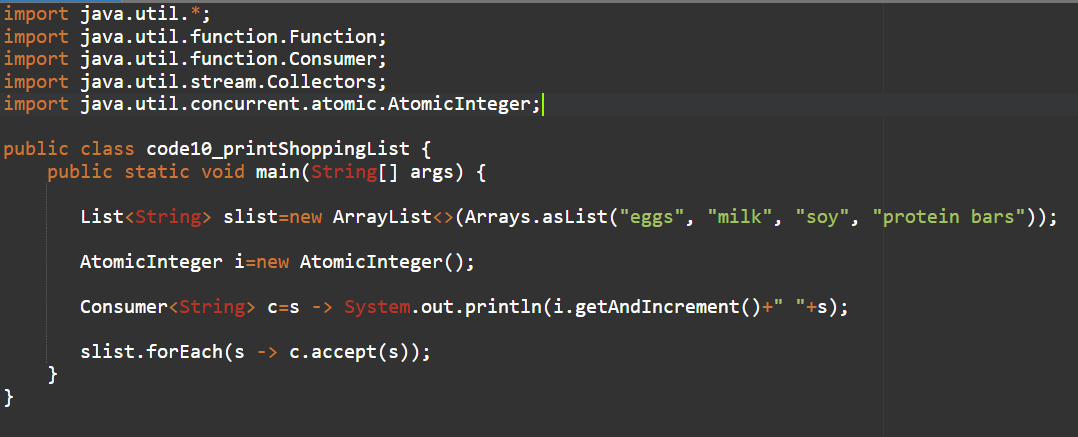
1. **Parse a date string "2025-08-08" into a LocalDate.**

****

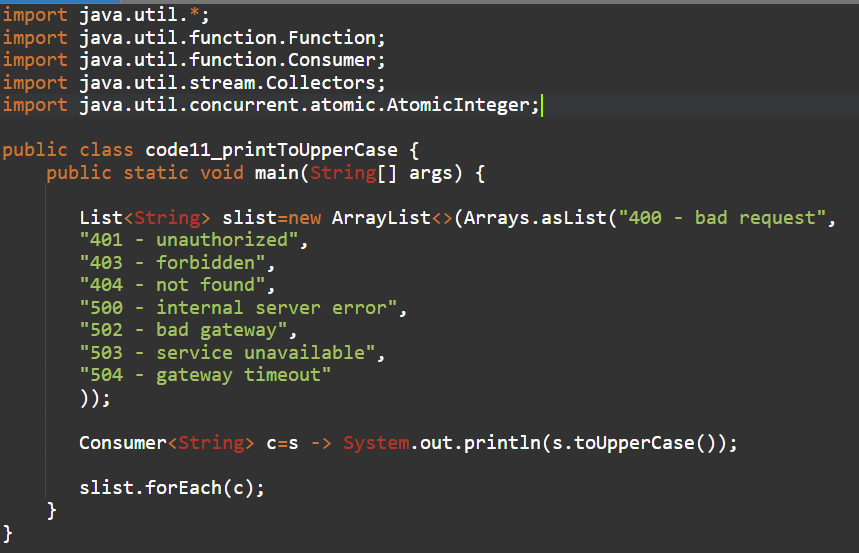
1. **Convert Celsius temperatures to Fahrenheit.**

****

1. **Print each item in a shopping list with numbering.**

****

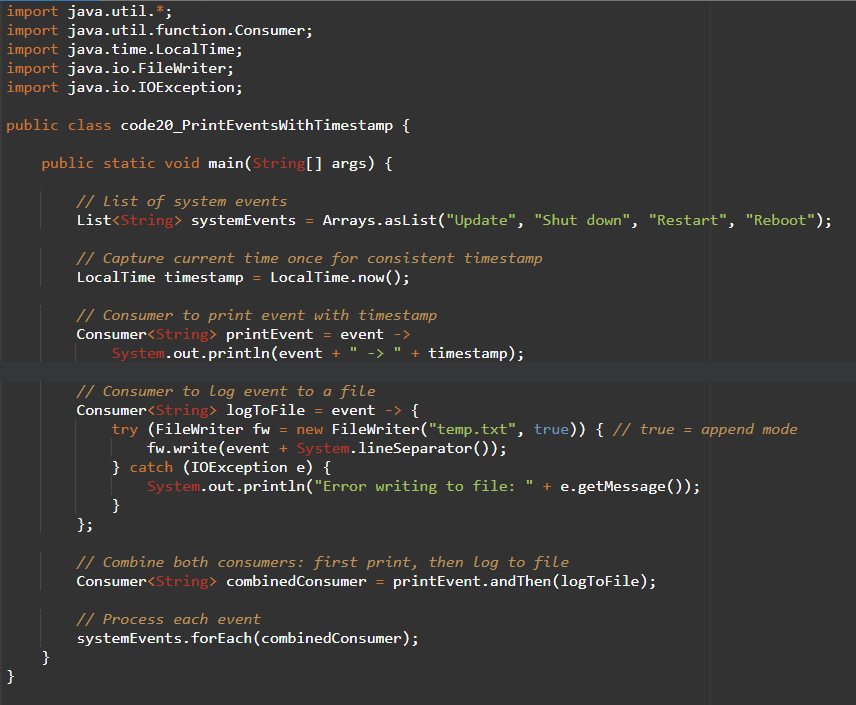
1. **Log all error messages in uppercase.**

****

1. **Logging System Events**

**Write a program that takes a list of system events (strings) and uses a Consumer<String> to:**

* **Print each event with a timestamp.**
* **Log it to a file simultaneously. *(Hint: chain Consumers with andThen())***

****

1. **Order Processing Pipeline**

**Given a list of orders (objects with id, item, price), use a Consumer<Order> to:**

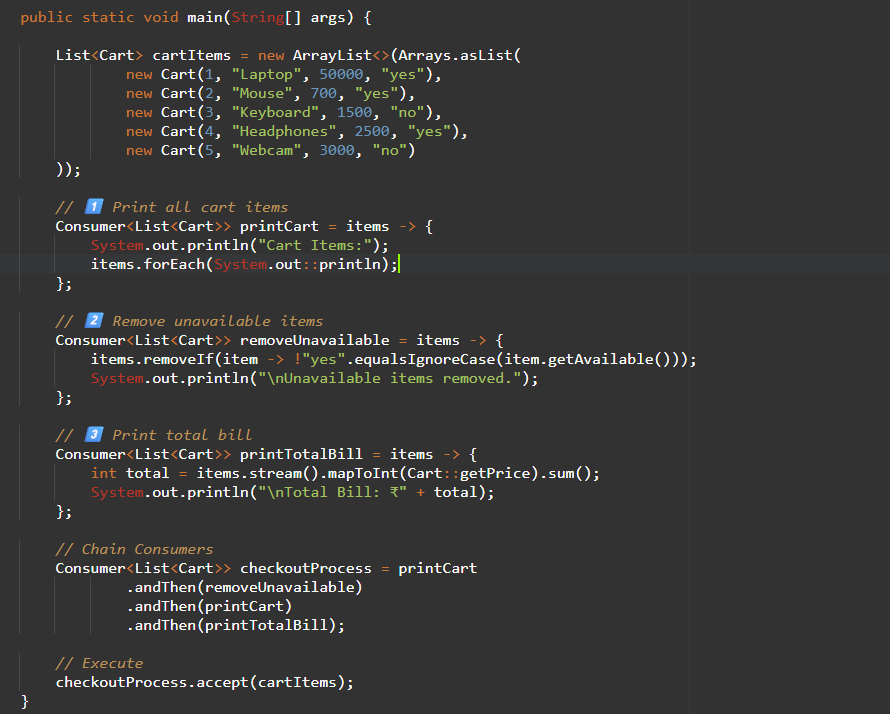
* **Print order details.**
* **Apply a discount if the price > 1000.**
* **Send a confirmation message. *(Hint: multiple Consumers chained)***



1. **Shopping Cart Checkout**

Simulate a shopping cart checkout process where Consumer<List<String>> is used to:

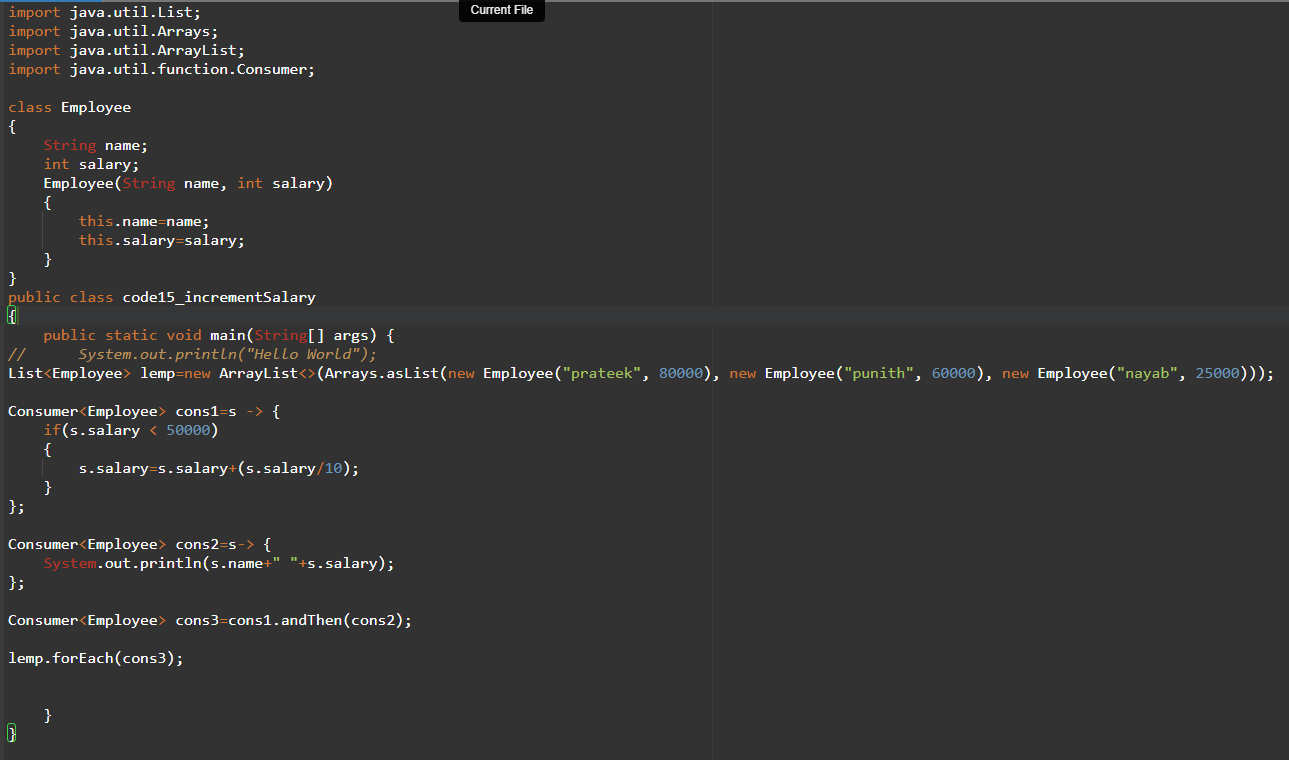
* Print all items in cart.
* Remove unavailable items.
* Print total bill after modifications.



1. **Employee Salary Update**

Given a list of employees (name, salary), use a Consumer<Employee> to:

* Increase salary by 10% if salary < 50,000.
* Print updated employee details.



1. **Chained Notification System**

Write a program where Consumer<String>:

* Sends a notification to the console.
* Writes the notification to a log file.
* Sends the notification to an API endpoint (simulate HTTP POST).

import java.util.List;

import java.util.Arrays;

import java.util.ArrayList;

import java.util.function.Consumer;

import java.time.LocalDateTime;

import java.io.FileWriter;

import java.io.IOException;

import java.net.URI;

import java.net.http.HttpClient;

import java.net.http.HttpRequest;

import java.net.http.HttpResponse;

public class code16\_sendNotificationToServer

{

public static void main(String[] args) {

LocalDateTime lt=LocalDateTime.now();

Consumer<String> cons1=s->System.out.println(s+" "+lt);

Consumer<String> cons2=s->

{

try(FileWriter fw=new FileWriter("log.txt", true))

{

fw.write(s+" "+System.lineSeparator());

}

catch(IOException e)

{

System.out.print(e.getMessage());

}

};

Consumer<String> cons3=s->

{

try

{

HttpClient client = HttpClient.newHttpClient();

HttpRequest hreq=HttpRequest.newBuilder()

.uri(URI.create("https://httpbin.org/post")) // Test API

.header("Content-Type", "application/json")

.POST(HttpRequest.BodyPublishers.ofString("{\"notification\":\"" + s + "\"}"))

.build();

HttpResponse hres=client.send(hreq, HttpResponse.BodyHandlers.ofString());

System.out.println(hres.statusCode());

System.out.println(hres.body());

}

catch(Exception e)

{

System.out.print(e.getMessage());

}

};

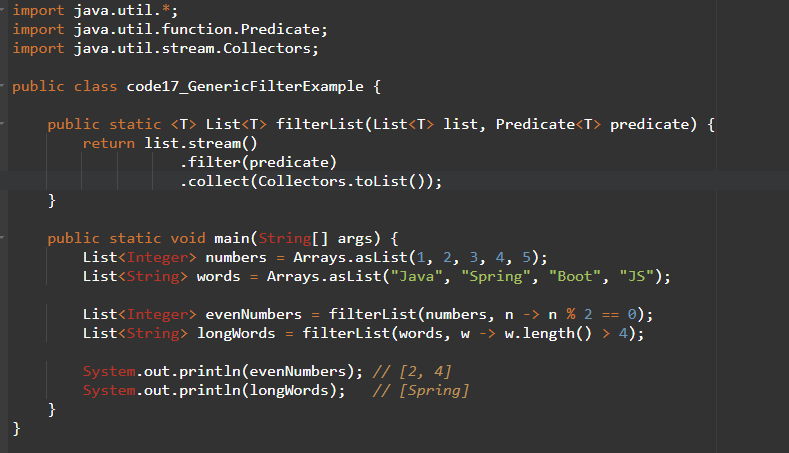
Consumer<String> cons4=cons1.andThen(cons2).andThen(cons3);

cons4.accept("hey prateek");

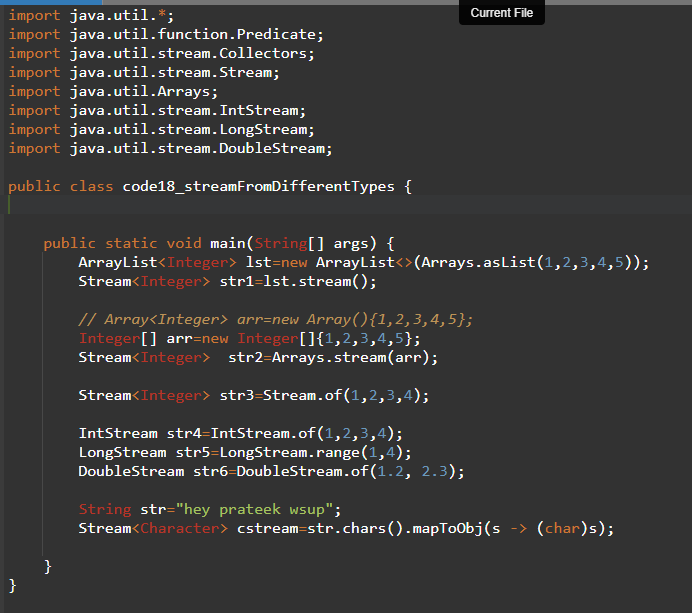
}

}

1. **Create a generic method that accepts a Predicate<T> to filter any list type.**

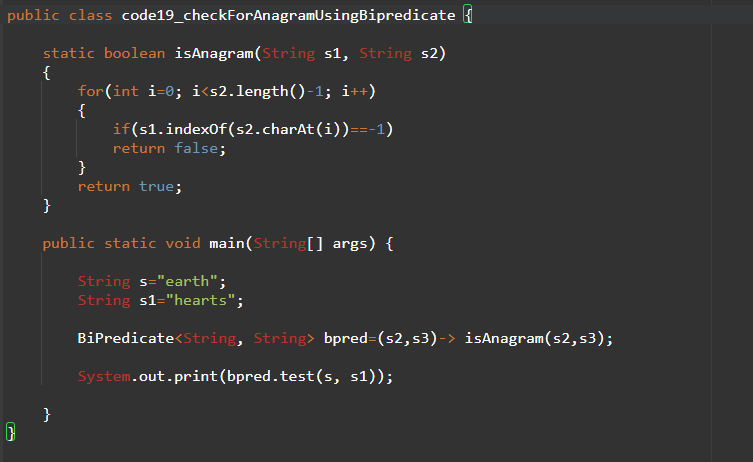
****

1. **Stream from different types**

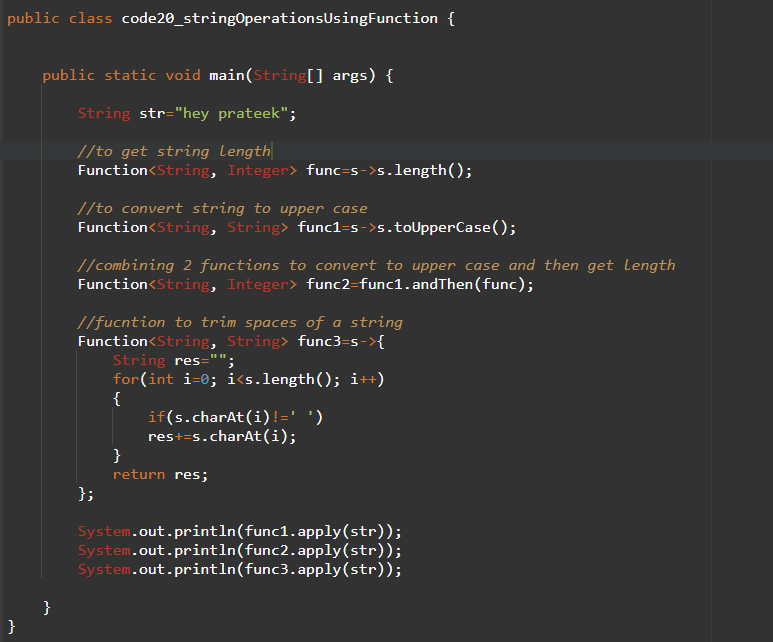
****

**BiPredicate**

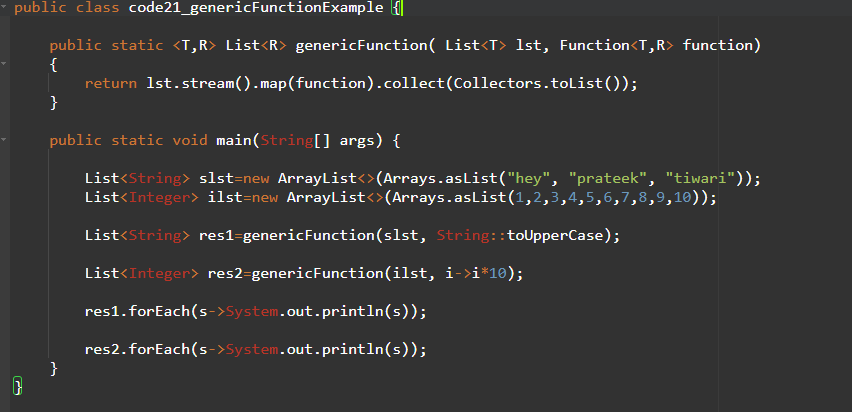
1. **Check if two strings are anagrams.**



1. **String operations using Function**



1. **Create a generic method that applies a given Function to each list element.**



1. **Write a generic method that takes a BiFunction for merging any two lists.**