1. Write a program to calculate the Simple Interest with minimal code using features of Java 11.

package io.Java;

import java.util.Scanner;

public class SimpleInterest {

public static void main(String[] args) {

Scanner s = new Scanner(System.in);

System.out.println("Enter the Amount, rate and time: ");

int p = s.nextInt();

int rate = s.nextInt();

int time = s.nextInt();

Interest i = (int a, int b, int c) -> (a\*b\*c)/100;

int ans = i.calculate(p, rate, time);

System.out.println(ans);

}

}

interface Interest

{

int calculate(int p, int rate, int time);

}

Output:

Enter the Amount, rate and time:

100000

7

13

91000

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Java 11 supports var keyword for variable declarations. List the scenarios where var keyword cannot be used for such variable declarations, Give reason in support of your answer for each scenario.

* Scenario 1: var keyword cannot be used as an instance and a global variable

class Demo3 {

// instance variable

var x = 50;

public static void main(String[] args)

{

System.out.println(x);

}

}

Output:

prog.java:8: error: 'var' is not allowed here

var x = 50;

^

1 error

* Scenario 2: var cannot be used as a generic type

import java.util.\*;

class Demo4 {

public static void main(String[] args)

{

// Generic list using var

var<var> al = new ArrayList<>();

// add elements

al.add(10);

al.add(20);

al.add(30);

// print the list

System.out.println(al);

}

}

Output:

prog.java:10: error: 'var' is not allowed here

var<var> al = new ArrayList<>();

^

1 error

* Scenario 3: var cannot be used without explicit initilization

import java.io.\*;

class Demo6 {

public static void main(String[] args)

{

// declaration without

// initialization

var variable;

// This is also not valid

var variable = null;

}

}

Output:

prog.java:13: error: cannot infer type for local variable variable

var variable;

^

(cannot use 'var' on variable without initializer)

prog.java:16: error: cannot infer type for local variable variable

var variable = null;

^

(variable initializer is 'null')

2 errors

* Scenario 4: var cannot be used with lambda expression

import java.util.\*;

interface myInt {

int add(int a, int b);

}

class Demo7 {

public static void main(String[] args)

{

// var cannot be used since they

// require explicit target type

var obj = (a, b) -> (a + b);

// calling add method

System.out.println(obj.add(2, 3));

}

}

Output:

prog.java:13: error: cannot infer type for local variable obj

var obj = (a, b) -> {

^

(lambda expression needs an explicit target-type)

1 error

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. A quick brown fox jumps over the lazy dog". Create an ArrayList from the given String. Such an ArrayList should include each word from the given sentence. Finally. convert such List to an array using Java 11 methods and print the output.

package io.sss;

import java.util.ArrayList;

import java.util.Arrays;

import java.util.List;

public class StringOp {

public static void main(String[] args)

{

String str = "A quick brown fox jumps over the lazy dog";

String words[] = str.split(" ");

List<String> l = new ArrayList<String>();

for(String text:words) {

l.add(text);

}

System.out.println(l);

//java11

String[] array = l.toArray(String[]::new);

System.out.println("In java 11 features : " + Arrays.toString(array));

}

}

Output:

[A, quick, brown, fox, jumps, over, the, lazy, dog]

In java 11 features : [A, quick, brown, fox, jumps, over, the, lazy, dog]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Using features of Java 11, read the data from a text file (File name: StudentList.bt). Calculate the count of students and print the names as well as the total count of students on the screen. (If any line in file doesn't contain a name, for such a record blank space should not be printed in the output)

* Hint: Use java 11 features of files and String methods to reduce the lines of code to be written.

package io.sss;

import java.io.IOException;

import java.nio.file.Files;

import java.nio.file.Path;

import java.nio.file.Paths;

public class FileClass {

public static void main(String[] args)

{

Path filePath = Paths.get("C:/","Users","SUBSATPA","Desktop","StudentList.txt");

try

{

String line;

int count =0;

String content = Files.readString(filePath);

System.out.println(content);

String words[] = content.split(" ");

count = count + words.length;

System.out.println("Total Name Count: " + count);

}

catch (IOException o)

{

o.printStackTrace();

}

}

}

Output:

Harshita Lutika Niharika Subhasish Vedant Rohan Subham

Total Name Count: 7

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Write a code using HttpClient API which sends a GET request to <https://httpbin.org/get>, and print out the response header, status code, and body for the given URL Sample output could be (Note: date and other attribute values may differ in your case.

package io.sss;

import java.io.IOException;

import java.net.URI;

import java.net.http.HttpClient;

import java.net.http.HttpRequest;

import java.net.http.HttpResponse;

import java.net.http.HttpClient.Version;

import java.net.http.HttpResponse.BodyHandlers;

public class URIDemo {

public static void main(String[] args) {

String uri = "https://httpbin.org/get";

HttpRequest req = HttpRequest.newBuilder()

.uri(URI.create(uri))

.GET()

.version(Version.HTTP\_2)

.build();

HttpClient client = HttpClient.newBuilder()

.build();

try

{

HttpResponse<String> resp = client.send(req, BodyHandlers.ofString());

System.out.println(resp.headers());

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

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

}

catch(IOException | InterruptedException e)

{

e.printStackTrace();

}

}

}

Output:

java.net.http.HttpHeaders@ae0d7a0b { {:status=[200], access-control-allow-credentials=[true], access-control-allow-origin=[\*], content-length=[245], content-type=[application/json], date=[Thu, 20 Jan 2022 14:57:38 GMT], server=[gunicorn/19.9.0]} }

200

{

"args": {},

"headers": {

"Host": "httpbin.org",

"User-Agent": "Java-http-client/17.0.1",

"X-Amzn-Trace-Id": "Root=1-61e97862-5de22f4c7cf96ecf7ac62a5f"

},

"origin": "117.217.49.63",

"url": "https://httpbin.org/get"

}