**Java 11 Assignment**

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

Hint Use the concept of functional interface and Local variable syntax for lambda parameters.

**package** io.skjava11;

**import** java.util.Scanner;

**public** **class** SimpleIntrest {

**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:

500

10

5

250

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 Test1 {

// instance variable

var a = 12;

public static void main(String[] args)

{

System.out.println(a);

}

}

OUTPUT:

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

var a = 12;

^

1 error

Scenario 2: var cannot be used as a generic type

import java.util.\*;

class Test2 {

public static void main(String[] args)

{

// Generic list using var

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

// add elements

list.add(14);

list.add(25);

list.add(43);

// print the list

System.out.println(list);

}

}

OUTPUT:

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

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

^

1 error

Scenario 3: var cannot be used without explicit initialization

import java.io.\*;

class Test3 {

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 AirthM {

int add(int x, int y);

}

class Test4 {

public static void main(String[] args)

{

// var cannot be used since they

// require explicit target type

var obj = (x, y) -> (x + y);

// calling add method

System.out.println(obj.add(12, 7));

}

}

OUTPUT:

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

var obj = (x, y) -> {

^

(lambda expression needs an explicit target-type)

1 error

3.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.sk;

**import** java.util.ArrayList;

**import** java.util.Arrays;

**import** java.util.List;

**public** **class** StringOperation {

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

{

String str = "Change the word by being yourself";

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

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

**for**(String text:words) {

list.add(text);

}

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

//java11

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

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

}

}

OUTPUT:

[Change, the, word, by, being, yourself]

In java 11 features : [Change, the, word, by, being, yourself]

4.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.sk;

**import** java.io.IOException;

**import** java.nio.file.Files;

**import** java.nio.file.Path;

**import** java.nio.file.Paths;

**public** **class** Count {

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

{

Path filePath = Paths.*get*("C:\"Users","SUSHEKUM","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:

Susheel Ajay Suresh Jay Manish

Total Name Count: 5

5.Write a program with the menu to accept the price of certain items and display their total. When user selects Option 1: should accept the prices of different products and insert these prices into first file (each amount to be inserted in a newline in the file). Next total of these values should be saved in a new file. Option 2: should allow the user to view the total of these prices from the second file.

Select your option (1: Insert New Price, 2: View Purchase Total, 3: Exit)

>1

>Insert 1st price:

>100

>Price has been saved to the file

>Do you want to enter price for more items? (Yes/No)

>Yes

>Insert 2nd price:

>200

>Price has been saved to the file

>Do you want to enter price for more items? (Yes/No)

>No

> Select your option (1: Insert New Price, 2: View Purchase Total, 3: Exit)

>2

>Total Price of all items is: 300

> Select your option (1: Insert New Price, 2: View Purchase Total, 3: Exit)

>3

exit program.....

HINT: Use java 11 features of files and String methods to reduce the line of code.

6.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.sk;

**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** URITest {

**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@ae7db49c { {:status=[200], access-control-allow-credentials=[true], access-control-allow-origin=[\*], content-length=[244], content-type=[application/json], date=[Fri, 21 Jan 2022 03:26:40 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=1-61ea27f0-25b865202b37737407cd3832"

},

"origin": "47.9.191.147",

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

}