**Assignments on Java 11**

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

HINT: Use the concepts of functional interface and Local variable syntax for lambda parameters.

**import** java.util.Scanner;

@FunctionalInterface

**interface** interest{

**int** formula(**int** a,**int** b, **int** c);

}

**public** **class** SI {

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

// **TODO** Auto-generated method stub

**int** p,r,t;

Scanner sc = **new** Scanner(System.***in***);

System.***out***.println("Enter the principal: ");

p = sc.nextInt();

System.***out***.println("Enter the rate: ");

r = sc.nextInt();

System.***out***.println("Enter the time: ");

t = sc.nextInt();

interest si = (a, b, c) -> (a\*b\*c)/100;

**int** ans = si.formula(p,r,t);

System.***out***.println("Simple Interest: " +ans);

}

}

**Output:** Enter the principal:

70000

Enter the rate:

5

Enter the time:

2

Simple Interest: 7000

1. Java 11 supports **var** keyword for variable declarations. List the scenarios where **var**

Keyword cannot be used for such variable declaration. Give reason in support of your answer for each scenario.

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

**package** org;

**class** demo1 {

**var** x = 50;

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

// **TODO** Auto-generated method stub

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

}

}

**Output:** Cannot make a static reference to the non-static field x

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

**package** org;

**import** java.util.\*;

**class** demo2 {

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

// **TODO** Auto-generated method stub

**var**<**var**> v = **new** ArrayList<>();

// add elements

v.add(10);

v.add(20);

v.add(30);

// print the list

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

}

}

**Output:** Cannot prog.java:10:error: ‘var’ is not allowed here

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

1 error

|  |
| --- |
|  |
|  |  |  |
|  |  | ^ |
|  |  |  |

* Scenario 3: var cannot be used without explicit initialization

**package** org;

**import** java.io.\*;

**class** demo3 {

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

// **TODO** Auto-generated method stub

// declaration without

// initialization

**var** variable;

// This is also not valid

**var** variable = **null**;

}

}

**Output:** Cannot use 'var' on variable without initializer

Duplicate local variable variable

Cannot infer type for local variable initialized to 'null'

* Scenario 4: var cannot be used with lambda expression

**package** org;

**interface** myInt {

**int** add(**int** a, **int** b);

}

**class** demo4 {

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

// **TODO** Auto-generated method stub

**var** v = (a,b) -> (a + b);

System.***out***.println(v.add(2,3));

}

}

**Output:** Lambda expression needs an explicit target-type

Cannot invoke add(int, int) on the primitive type null

1. “A quick brown for jumps over the last 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** org;

**import** java.util.ArrayList;

**import** java.util.Arrays;

**import** java.util.List;

**public** **class** sentence {

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

// **TODO** Auto-generated method stub

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

String words[] = st.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("Java 11 features : " + Arrays.*toString*(array));

}

}

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

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

1. Using features of java 11, read the data from text file (**File name:** StudentList.txt). Calculate the count of students and print the names as well as the total count of students on the screen.(If any line in the 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** org;

**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) {

// **TODO** Auto-generated method stub

Path filePath = Paths.*get*("C:/","Users","TEJASWI","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:** Tej Sai gayee Bhargav sony

Total Name Count: 5

1. 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.

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 org;

import java.io.IOException;

import java.net.URI;

import java.net.URL;

import java.net.URLConnection;

import java.net.http.HttpClient;

import java.net.http.HttpRequest;

import java.net.http.HttpResponse;

import java.util.List;

import java.util.Map;

public class http\_uri {

public static void main(String[] args) throws IOException, InterruptedException {

// TODO Auto-generated method stub

HttpClient client = HttpClient.newHttpClient();

HttpRequest request = HttpRequest.newBuilder()

.uri(URI.create("http://httpbin.org/get"))

.GET() // GET is default

.build();

HttpResponse<Void> response = client.send(request,

HttpResponse.BodyHandlers.discarding());

System.out.println("Status code is: "+response.statusCode());

try {

URL obj = new URL("http://httpbin.org/get");

URLConnection conn = obj.openConnection();

Map<String, List<String>> map = conn.getHeaderFields();

System.out.println("Printing Response Header...\n");

for (Map.Entry<String, List<String>> entry : map.entrySet()) {

System.out.println("Key : " + entry.getKey()

+ " ,Value : " + entry.getValue());

}

System.out.println("\nGet Response Header By Key ...\n");

String server = conn.getHeaderField("Server");

if (server == null) {

System.out.println("Key 'Server' is not found!");

} else {

System.out.println("Server - " + server);

}

System.out.println("\n Done");

} catch (Exception e) {

e.printStackTrace();

}

}

}

**Ouput:**  Status code is: 200

Printing Response Header...

Key : null ,Value : [HTTP/1.1 200 OK]

Key : Server ,Value : [gunicorn/19.9.0]

Key : Access-Control-Allow-Origin ,Value : [\*]

Key : Access-Control-Allow-Credentials ,Value : [true]

Key : Connection ,Value : [keep-alive]

Key : Content-Length ,Value : [302]

Key : Date ,Value : [Fri, 21 Jan 2022 05:48:08 GMT]

Key : Content-Type ,Value : [application/json]

Get Response Header By Key ...

Server - gunicorn/19.9.0

Done