Core Java Assignment 10

1.

**public** **class** SimpleInterest {

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

**int** a=100;

**int** b=10;

**int** c=2;

Simplei si=(p,t,r)->(a\*b\*c/100);

System.out.println(si);

}

@FunctionalInterface

**interface** Simplei{

**int** Simple(**int** p,**int** t,**int** r);

}

}

2.

//Scenario 1-var cannot be used in an instance and global variable declaration

//as we Cannot make a static reference to the non-static field x

/\*class Demo1 {

var x = 50; // instance variable

public static void main(String[] args)

{

System.out.println(x);

}

}

\*/

/\* Scenario 2-var cannot be used as a Generic type. as it cannot be resolved into perticular type

class Demo2 {

public static void main(String[] args)

{

var<var> al = new ArrayList<>(); // Generic list using var

al.add(10); // add elements

al.add(20);

al.add(30);

System.out.println(al); // print the list

}

}

\*/

//Scenario 3-var cannot be used without explicit initialization.

/\*as Cannot use 'var' on variable without initializer

Duplicate local variable variable Cannot infer type for local variable initialized to 'null'

class Demo3 {

public static void main(String[] args)

{

var variable; // Declaration without initialization

var variable = null; // This is also not valid

}

}

\*/

/\*Scenario 4- var cannot be used with Lambda Expression

\* var cannot be used since they require explicit target type

\* interface myInt {

int add(int a, int b);

}

class Demo4 {

public static void main(String[] args)

{

var obj = (a, b) -> (a + b); // var cannot be used since they require explicit target type

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

}

}

//Scenario 5-var cannot be used for method parameters and return type

\* class Demo8 {

// method1 using var

// as a return type

var method1() { return ("Inside Method1"); }

// method2 using var for a

// parameter

void method2(var a) { System.out.println(a); }

public static void main(String[] args)

{

Demo1 obj = new Demo1(); // create an instance

var res = obj.method1(); // call method1

obj.method2(); // call method2

}

}

\*/

public class Assignment10\_Java11\_Q2\_var {

public static void main(String[] args) {

}

}

3.

import java.util.ArrayList;

import java.util.List;

import java.util.Arrays;

public class StringtoArray {

public static void main(String[] args) {

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

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

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

al = Arrays.asList(str);

for(String s: al){

System.out.println(s);

}

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

String[] aar=al.stream().toArray(size->new String[size]);

System.out.println(Arrays.toString(aar));

}

}

4.

import java.io.BufferedReader;

import java.io.File;

import java.io.FileNotFoundException;

import java.io.FileReader;

import java.io.IOException;

import java.nio.file.Files;

public class Assignment10\_Java11\_Q4 {

public static void main(String[] args) {

var path="C:\\Users\\shree\\Desktop\\Core\_Java\_Assignments\\Student\_List.txt";

File file = new File("C:\\Users\\shree\\Desktop\\Core\_Java\_Assignments\\Student\_List.txt");

try {

BufferedReader br = new BufferedReader(new FileReader(file));

String line;

System.out.println("Reading Data From file: ");

while ((line = br.readLine()) != null)

System.out.println(line);

System.out.println("\nSpaces removing:");

Files.lines(new File("C:\\Users\\shree\\Desktop\\Core\_Java\_Assignments\\Student\_List.txt").toPath())

.map(s -> s.trim()).filter(s -> !s.isEmpty()).forEach(System.out::println);

System.out.println("\nTotal count of students: ");

Long count= Files.lines(new File("C:\\Users\\shree\\Desktop\\Core\_Java\_Assignments\\Student\_List.txt").toPath())

.map(s -> s.trim())

.filter(s -> !s.isEmpty()).count();

System.out.println(count);

} catch (FileNotFoundException e) {

e.printStackTrace();

} catch (IOException e) {

e.printStackTrace();

}

}

}

6.

import java.io.IOException;

import java.net.URI;

import java.net.http.HttpClient.Version;

import java.net.http.HttpClient;

import java.net.http.HttpRequest;

import java.net.http.HttpResponse;

import java.net.http.HttpResponse.BodyHandlers;

public class HTTP {

public static void main(String[] args) {

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

HttpRequest req=HttpRequest.newBuilder()

.uri(URI.create(url))

.GET()

.version(Version.HTTP\_2)

.build();

HttpClient client = HttpClient.newBuilder()

.build();

try {

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

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

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

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

} catch (IOException | InterruptedException e) {

e.printStackTrace();

}

}

}