**Assignments on Java 11**

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.

@FunctionalInterface

**interface** MyInterface{

// abstract method

**double** getPiValue();

}

**public** **class** first {

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

// declare a reference to MyInterface

MyInterface ref;

**float** P , R , T , SI ;

P =50; R =30; T = 5;

SI = (P\*R\*T)/100;

// lambda expression

ref = () ->SI;

System.***out***.println("Simple Interest= " + SI);

}

}

**OUTPUT:**

Simple Interest= 75.0

2. 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  It cannot be used for instance variables at class level

**package** Java11;

**public** **class** VarKeyword

{

// instance variable

**var** x = 50;

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

{

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

}

}

**Output:**

Exception in thread "main" java.lang.Error: Unresolved compilation problem:

Cannot make a static reference to the non-static field x

at Java11.VarKeyword.main(VarKeyword.java:9)

Scenario 2: var cannot be used as a generic type

**package** Java11;

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

**public** **class** VarKeyword

{

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

{

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

al.add(123);

al.add(456);

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

}

}

**Output:**

[123, 456]

Scenario 3: var cannot be used without explicit initialization

**package** Java11;

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

**class** VarKeyword {

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

**var** variable;

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

}

}

**Output:**

Exception in thread "main" java.lang.Error: Unresolved compilation problems:

Cannot use 'var' on variable without initializer

Duplicate local variable variable

Cannot infer type for local variable initialized to 'null'

at Java11.VarKeyword.main(VarKeyword.java:8)

Scenario 4: var cannot be used with lambda expression

**package** Java11;

**interface** Integer {

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

}

**class** VarKeyword {

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

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

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

}

}

**Output:**

Exception in thread "main" java.lang.Error: Unresolved compilation problems:

Lambda expression needs an explicit target-type

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

at Java11.VarKeyword.main(VarKeyword.java:9)

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** javaEleven;

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

**class** abc

{

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

{

String[] str = {"A", "quick" ,"brown"," fox" ,"jumps" ,"over","the"," lazy" ,"dog"};

List list = Arrays.*asList*(str);

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

}

}

**OUTPUT:**

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

4. Using features of Java 11, read the data from a 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

**package** javaEleven;

**import** java.io.File;

**import** java.io.IOException;

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

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

**import** java.util.Scanner;

**class** fourth

{

**public** **static** String readFile(String path) **throws** IOException {

**return** Files.*readString*(Paths.*get*(path));

}

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

{

String filePath = "C:\\Users\\hp\\Desktop\\GOT\\StudentList.txt";

String content = **null**;

**try** {

content = *readFile*(filePath);

} **catch** (IOException e) {

e.printStackTrace();

}

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

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

**int** count = 0;

**try** {

// create a new file object

File file = **new** File("C:\\\\Users\\\\akcha\\\\Desktop\\\\GOT\\\\StudentList.txt");

Scanner sc = **new** Scanner(file);

**while**(sc.hasNextLine())

{

sc.nextLine();

count++;

}

System.***out***.println("Total Number of Students: " + count);

sc.close();

} **catch** (Exception e) {

e.getStackTrace();

}

}

}

**OUTPUT:**

Shubham

Akshat

Tanay

Sourabh

Ashish

Ankit

Ravi

Ayush

Saharsh

Total Number of Students: 9

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.

**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** httpiAPI {

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

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();

}

}

}

**OUTPUT:**

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 : [305]

Key : Date ,Value : [Thu, 20 Jan 2022 14:06:52 GMT]

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

Get Response Header By Key ...

Server - gunicorn/19.9.0

Done