Submitted by – Sanket Bollamwar

Assignments on Java 11

Q1. 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.

**Code:**

interface SimpleInterest{

    public double Si(int principal,int time,double rate);

}

public class Assignment8Q1 {

    public static void main(String[] args) {

        SimpleInterest simpleInterest = ((principal, time, rate) -> (principal\*time\*rate)/100 );

        System.out.println("Simple Interest\nPrincipal = 10000\nRate of interest = 10\nTime = 6");

        System.out.println("Simple interest = "+simpleInterest.Si(10000,6,10));

        //not required

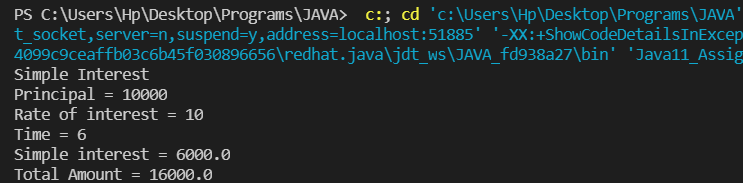
        double amount = 10000+simpleInterest.Si(10000,6,10);

        System.out.println("Total Amount = "+amount);

    }

}

**Output:**

****

Q2. 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.

**Answer:**

* You can use var only for local variables (in methods). It cannot be used for instance variables (at class level).
* You cannot use var in Lambda expressions.
* You cannot use var for method signatures (in return types and parameters).

And remember, you cannot use var to declare a variable without explicit initialization, hence the following:

**var x;**

is not allowed, since local variable declaration requires initialization on the right side. That also means this declaration is not valid:

**var x = null;**

Q3. “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.

**Code:**

import java.util.ArrayList;

//A quick brown fox jumps over the lazy dog.

public class Assignment8Q3 {

    public static void main(String[] args) {

        ArrayList<String> arrayList = new ArrayList<String>();

        arrayList.add("A");

        arrayList.add("quick");

        arrayList.add("brown");

        arrayList.add("fox");

        arrayList.add("jumps");

        arrayList.add("over");

        arrayList.add("the");

        arrayList.add("lazy");

        arrayList.add("dog");

        //converting array list to array

        Object[] objects = arrayList.toArray();

        for (Object obj: objects){

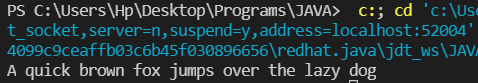
            System.out.print(obj+" ");

        }

    }

}

**Output:**

****

Q4. 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. (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.

**Code:**

import java.io.\*;

import java.util.Scanner;

public class Assignment8Q4 {

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

        // File path is passed as parameter

        File file = new File("C:\\Users\\Hp\\Desktop\\Assignments\\Core Java 11\\Java11Assignments\_StudentList.txt");

        Scanner sc = new Scanner(file);

        int count = 0;

        while (sc.hasNextLine()){

            String string = sc.nextLine();

            if(string.length()!=0){

                System.out.println(string.trim());

                count++;

            }

        }

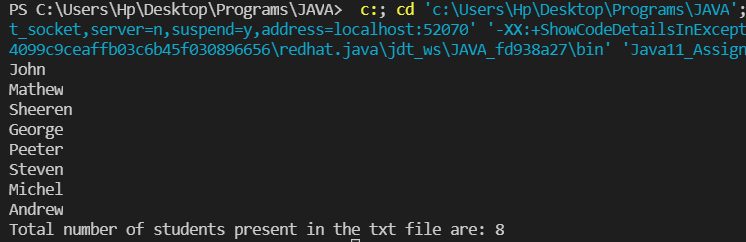
        System.out.println("Total number of students present in the txt file are: "+count);

        sc.close();

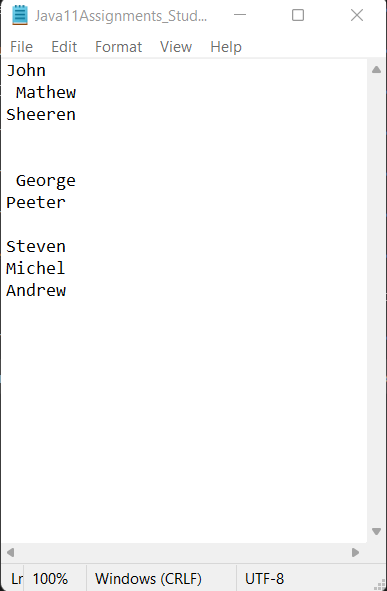
    }

}

**Output:**

****

**File:**

****

Q5. Write a program with 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.

**Sample Output**:

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.

**Code:**

import java.util.Locale;

import java.util.Scanner;

public class Assignment8Q5 {

    public static String ordinal(int i) {

        String[] suffixes = new String[]{"th", "st", "nd", "rd", "th", "th", "th", "th", "th", "th"};

        switch (i % 100) {

            case 11:

            case 12:

            case 13:

                return i + "th";

            default:

                return i + suffixes[i % 10];

        }

    }

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        int count = 0;

        int price,total = 0;

        int choice;

        do{

            System.out.println("1) INSERT NEW PRICE");

            System.out.println("2) VIEW PURCHASE TOTAL");

            System.out.println("3) EXIT");

            System.out.print(" \n ENTER YOUR CHOICE: ");

            choice = sc.nextInt();

            switch (choice){

                case 1:

                    String s = "Yes";

                    do{

                        System.out.print("Insert "+ordinal(count+1)+" price: ");

                        price = sc.nextInt();

                        total += price;

                        System.out.println("Price has been saved to the file");

                        System.out.print("Do you want to enter price for more items? (Yes/No): ");

                        s = sc.next();

                        count++;

                        System.out.println();

                    }while(s.equalsIgnoreCase("Yes"));

                    break;

                case 2:

                    System.out.println("Total Price of all items is: "+total);

                    break;

                case 3: System.exit(0);

                default : System.out.println("PLEASE ENTER THE CORRECT CHOICE!");

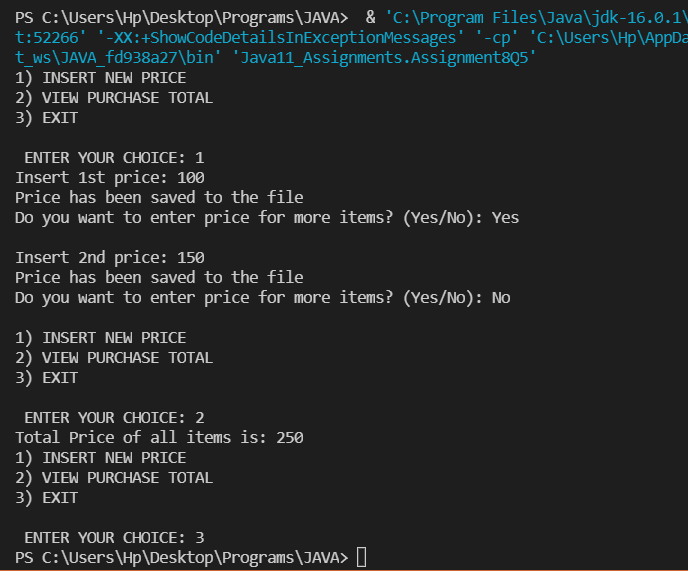
            }

        }while(choice!=3);

    }

}

**Output:**



Q6. 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 results):

access-control-allow-credentials:[true] access-control-allow-origin:[\*] connection:[keep-alive]

content-length:[273]

content-type:[application/json] date:[Fri, 06 Aug 2021 13:07:41 GMT]

server:[gunicorn/19.9.0] 200

{

"args": {}, "headers": {

"Content-Length": "0",

"Host": "httpbin.org",

"User-Agent": "Java 11 HttpClient Bot",

"X-Amzn-Trace-Id": "Root=1-610d341d-092dc33f698b192a219426d1"

},

"origin": "43.255.221.184",

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

}

**Code:**

package Java11\_Assignments;

import java.io.\*;

import java.net.\*;

import java.util.List;

import java.util.Map;

public class Assignment8Q6 {

    public static void main(String[] args)

    {

        try

        {

            URL url = new URL("https://httpbin.org/get");

            //open the connection to the above URL.

            HttpURLConnection http = (HttpURLConnection)url.openConnection();

            URLConnection urlcon = url.openConnection();

            Map<String, List<String>> header = urlcon.getHeaderFields();

            //print all the fields along with their value.

            for (Map.Entry<String, List<String>> mp : header.entrySet())

            {

                System.out.print(mp.getKey() + " : ");

                System.out.println(mp.getValue().toString());

            }

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

            List<String> contentLength = header.get("Content-Length");

            if (contentLength == null) {

                System.out.println("'Content-Length' doesn't present in Header!");

            } else {

                for (String header1 : contentLength) {

                    System.out.println("Content-Lenght: " + header1);

                }

            }

            int statusCode = http.getResponseCode();

            System.out.println("Status Code: "+statusCode);

            System.out.println();

            System.out.println("Complete source code of the URL is-");

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

            //get the inputstream of the open connection.

            BufferedReader br = new BufferedReader(new InputStreamReader

                    (urlcon.getInputStream()));

            String i;

            //print the source code line by line.

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

            {

                System.out.println(i);

            }

        }

        catch (Exception e)

        {

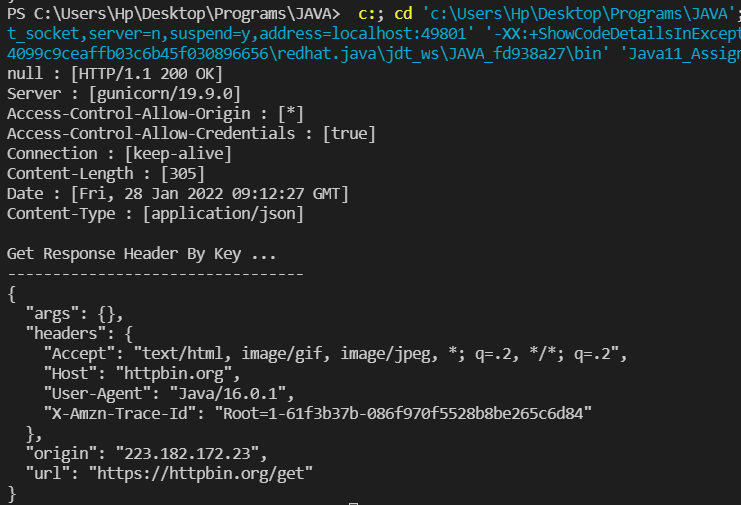
            System.out.println(e);

        }

    }

}

**Output:**

****