SUBJECT: OBJECT ORIENTED PROGRAMMING

Submitted To: Engr. Asmatullah

Submitted By: Asadullah Samo (21SW036)

Dated: 18-09-2022

Lab: Lab-10 Tasks

**Question 01:**

Write a java program based on filing. Perform Create, Read and Write operations on JSON file.

Source Code:

**package** Lab\_10\_Tasks;

**import** org.json.simple.JSONObject;

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

**import** java.io.IOException;

**public** **class** Task\_01 {

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

File myFile = **new** File("first.json");

// creating the File

**try** {

**if** (myFile.createNewFile())

System.out.println("File created "+myFile.getName());

}

**catch** (IOException ie){

System.out.println("File not found");

ie.printStackTrace();

}

// Writing in the file

JSONObject jsonObject = **new** JSONObject();

jsonObject.put("First\_Name", "Asad");

jsonObject.put("Caste", "Samo");

jsonObject.put("Age", "18");

jsonObject.put("Status", "Student");

**try**{

FileWriter file = **new** FileWriter("first.json");

file.write(jsonObject.toJSONString());

file.close();

}

**catch** (Exception e){

e.printStackTrace();

}

// Reading from the file

**try**{

Reader reader = **new** FileReader("first.json");

**int** data = reader.read();

**while**(data!=-1){

System.out.print((**char**)data);

data = reader.read();

}

reader.close();

}

**catch** (IOException ie){

System.out.println("File can't be read");

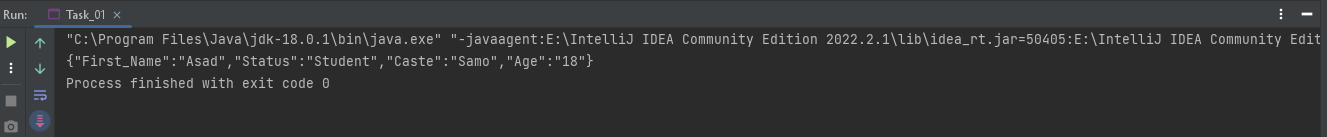
ie.printStackTrace();

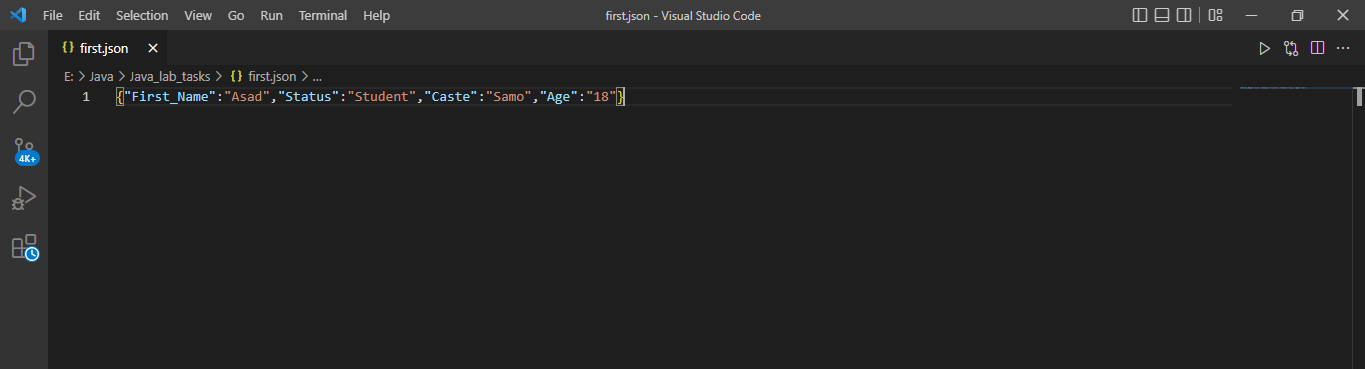
} // end of try catch block

} // end of main() method

}

OUTPUT:



****

**Question 02:**

Write a java program based on filing. Perform Create, Read and Write operations on text file using BufferedReader and BufferedWriter Class.

Source Code:

**package** Lab\_10\_Tasks;

**import** java.io.File;

**import** java.io.FileReader;

**import** java.io.FileWriter;

**import** java.io.BufferedReader;

**import** java.io.BufferedWriter;

**import** java.io.IOException;

**public** **class** Task\_02 {

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

// Creating a file

File file = **new** File("Second.txt");

**try** {

BufferedReader bf = **new** BufferedReader(**new** FileReader(file));

}

**catch** (IOException ie){

ie.printStackTrace();

} // end of try catch block

// Writing in the Second.txt

**try** {

FileWriter fw = **new** FileWriter("Second.txt");

BufferedWriter Write = **new** BufferedWriter(fw);

Write.write("Second.txt is written by bufferedWriter");

Write.close();

}

**catch** (IOException ie){

ie.printStackTrace();

}

// Reading from the Second.txt

**try**{

FileReader f = **new** FileReader("Second.txt");

BufferedReader br = **new** BufferedReader(f);

System.out.println(br.readLine());

}

**catch** (IOException ie){

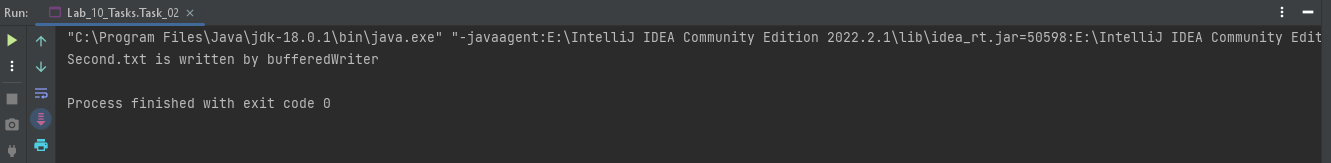
ie.printStackTrace();

}

} // end of main() method

} // end of program

OUTPUT:

****

**Graphical user interface, text, application, Word

Description automatically generated**

**Question 03:**

Write a program that opens a file in write mode(use file modes), takes name and roll no from the students write it in the file and closes the file.

Source Code:

**package** Lab\_10\_Tasks;

**import** java.io.File;

**import** java.io.FileReader;

**import** java.io.Reader;

**import** java.io.Writer;

**import** java.io.FileWriter;

**import** java.util.Scanner;

**public** **class** Task\_03 {

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

File file = **new** File("Third.txt");

file.setExecutable(**true**);

file.setReadable(**false**);

file.setWritable(**true**);

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

System.out.print("Enter your name: ");

String name = sc.nextLine();

System.out.print("Enter your roll number: ");

**int** roll\_no = sc.nextInt();

**try**{

Writer writer = **new** FileWriter("Third.txt");

writer.write(name+"\n"+roll\_no);

writer.close();

}

**catch** (Exception e){

e.printStackTrace();

}

**try**{

Reader reader = **new** FileReader("Third.txt");

**int** data = reader.read();

**while** (data!= -1){

System.out.print((**char**)data);

data = reader.read();

}

reader.close();

}

**catch** (Exception e){

e.printStackTrace();

}

} // end of main() method

} // end of program

OUTPUT:

**Text

Description automatically generated**

**Graphical user interface, application, Word

Description automatically generated**