Name: Gayathri S

EMP Id:46141877

**Project #1: Money Management**

Develop a personal finance application that helps you manage your money wisely.

The application will ask you to enter an amount you earn in this month (Account #0). This income is

divided into 6 sub-accounts which are described below:

• Account #1: **NEC** - Necessity - 55% of Account #0

• Account #2: **FFA** - Finance Freedom Account - 10% of Account #0

• Account #3: **EDU** - Education - 10% of Account #0

• Account #4: **LTSS** -Long Term Spending Saving - 10% of Account #0

• Account #5: **PLAY** - 10% of Account #0

• Account #6: **GIVE** - 5% of Account #0

Here’s an example:

Enter your income this month: 5000

Here’s how you should manage your money:

NEC: 2750 LTSS: 500

FFA: 500 PLAY: 500

EDU: 500 GIVE: 250

**NOTE**: Design the money dividing module in a way that is reusable in other applications, e.g.

desktop, web and mobile.

**SOURCE CODE :**

**package** salary;

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

**public** **class** Income {

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

**int** s;**double** a1,a2,a3,a4,a5,a6;

System.***out***.println("Enter ur Salary");

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

s=obj.nextInt();

a1 = (0.55\*s);

a2 = (0.1\*s);

a3 = (0.1\*s);

a4 = (0.1\*s);

a5 = (0.1\*s);

a6 = (0.05\*s);

System.***out***.println(" Manage Your Expenses like this");

System.***out***.println(" expense for NEC is:" + a1);

System.***out***.println(" expense for FFA is:" + a2);

System.***out***.println(" expense for EDU is:" + a3);

System.***out***.println(" expense for LTSS is:" + a4);

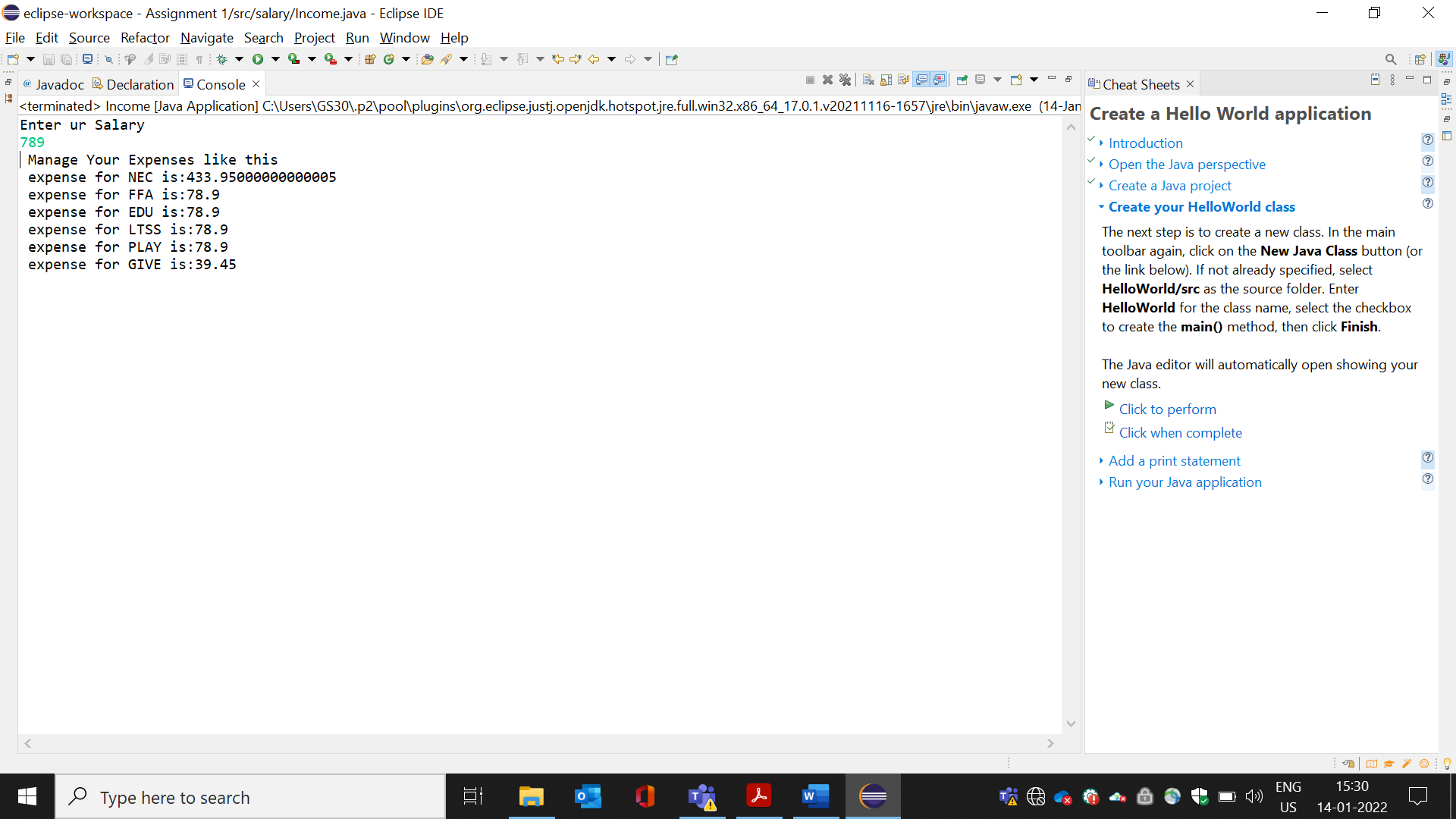
System.***out***.println(" expense for PLAY is:" + a5);

System.***out***.println(" expense for GIVE is:" + a6);

}

}

**OUTPUT :**



**Project #2: Human Resource Management**

Develop a simple Human Resources Management (HRM) application for an IT company. The

application manages the following kinds of employee:

- Programmer - Accountant - Team Leader - CTO

- Designer - Business Developer - Chairman - CFO

- Tester - Architect - CEO

You are required to develop the employee module which should be re-usable and extensible.

Here’s the required information for this module:

- Employee is the super type of all kinds of employees in the organization.

+ Attributes: name, email, address, birthday

+ Behaviors: work

- Programmer is a kind of employee:

+ Attributes: programming languages

+ Behaviors: code, fix bugs

- Tester is a kind of employee:

+ Attributes: testing methodologies.

+ Behaviors: test, verify.

- Designer is a kind of employee:

+ Attributes: design philosophy, design tools used

+ Behavior: design

- Business developer is a kind of employee:

+ Attributes: specialized domains

+ Behaviors: collect requirements, analyze requirements, write documents.

- Team leader is a kind of programmer:

+ Attributes: team collaboration tools, development methods

+ Behaviors: schedule, organize, manage programmers, designers and testers.

- Architect is a kind of programmer:

+ Attributes: design methodology

+ Behavior: design system

- CTO is a kind of architect who has:

+ Attributes: technology domains

+ Behaviors: advise technologies

- CEO is a kind of employee who has:

+ Attributes: vision, mission

+ Behavior: executive management

- CFO is a kind of employee who has:

+ Attributes: finance management skills

+ Behaviors: manage finance

You are required to implement entity classes for all these kinds of people using OOP features

abstraction, encapsulation and inheritance. Finally, package thelcompiled classes as a JAR file

(employee.jar) so it can be re-used among applications.

SOURCE CODE :

**package** com.capg.table;

**import** java.io.BufferedReader;

**import** java.io.IOException;

**import** java.io.InputStreamReader;

//import java.util.\*;

**class** Employee {

String name;String email;String address; String birthday;String work;String technology;

**public** **void** setWork(String work) {

**this**.work=work;

}

**public** **void** setTechnology(String technology) {

**this**.technology=technology;

}

**public** Employee(String name,String email,String address,String birthday) {

**this**.name = name;

**this**.email = email;

**this**.address = address;

**this**.birthday = birthday;

/\*this.technology = technology;

this.work = work;\*/

}

**public** **void** print() {

System.***out***.println("Name: "+name);

System.***out***.println("E-mail: "+email);

System.***out***.println("Address: "+address);

System.***out***.println("Birthday: "+birthday);

System.***out***.println("Technology: "+technology);

System.***out***.println("Work: "+work);

}

}

**class** Programmer **extends** Employee{

@Override

**public** **void** setWork(String work) {

**this**.work = work;

}

@Override

**public** **void** setTechnology(String technology) {

**this**.technology = technology;

}

**public** Programmer(String name,String email,String address,String birthday) {

**super**(name,email,address,birthday);

}

}

**class** Tester **extends** Employee{

@Override

**public** **void** setWork(String work) {

**this**.work = work;

}

@Override

**public** **void** setTechnology(String technology) {

**this**.technology = technology;

}

**public** Tester(String name,String email,String address,String birthday) {

**super**(name,email,address,birthday);

}

}

**class** Designer **extends** Employee{

@Override

**public** **void** setWork(String work) {

**this**.work = work;

}

@Override

**public** **void** setTechnology(String technology) {

**this**.technology = technology;

}

**public** Designer(String name,String email,String address,String birthday) {

**super**(name,email,address,birthday);

}

}

**class** Business\_Developer **extends** Employee{

@Override

**public** **void** setWork(String work) {

**this**.work = work;

}

**public** Business\_Developer(String name,String email,String address,String birthday) {

**super**(name,email,address,birthday);

}

}

**class** Team\_Leader **extends** Employee{

@Override

**public** **void** setWork(String work) {

**this**.work = work;

}

@Override

**public** **void** setTechnology(String technology) {

**this**.technology = technology;

}

**public** Team\_Leader(String name,String email,String address ,String birthday) {

**super**(name,email,address,birthday);

}

}

**class** Architect **extends** Employee{

@Override

**public** **void** setWork(String work) {

**this**.work = work;

}

@Override

**public** **void** setTechnology(String technology) {

**this**.technology = technology;

}

**public** Architect(String name,String email,String address,String birthday) {

**super**(name,email,address,birthday);

}

}

**class** CTO **extends** Architect{

@Override

**public** **void** setWork(String work) {

**this**.work = work;

}

@Override

**public** **void** setTechnology(String technology) {

**this**.technology = technology;

}

**public** CTO(String name,String email,String address, String birthday) {

**super**(name,email,address,birthday);

}

}

**class** CEO **extends** Employee{

@Override

**public** **void** setWork(String work) {

**this**.work = work;

}

@Override

**public** **void** setTechnology(String technology) {

**this**.technology = technology;

}

**public** CEO(String name,String email,String address,String birthday,String technology,String work) {

**super**(name,email,address,birthday);

}

}

**class** CFO **extends** Employee{

@Override

**public** **void** setWork(String work) {

**this**.work = work;

}

@Override

**public** **void** setTechnology(String technology) {

**this**.technology=technology;

}

**public** CFO(String name,String email,String address,String birthday) {

**super**(name,email,address,birthday);

}

}

**public** **class** HumanResource {

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

BufferedReader br = **new** BufferedReader(**new** InputStreamReader(System.***in***));

System.***out***.println("Enter Employee Designation: ");

String employee = br.readLine();

System.***out***.println("Enter Name: ");

String name = br.readLine();

System.***out***.println("Enter E-mail: ");

String email = br.readLine();

System.***out***.println("Enter Address: ");

String address = br.readLine();

System.***out***.println("Enter Birthday");

String birthday = br.readLine();

Employee e = **new** Employee(name,email,address,birthday);

**switch**(employee){

**case** "Programmer":

e.setTechnology("Programming");

e.setWork("code,fix bugs");

**break**;

**case** "Tester":

e.setTechnology("Testing Methodogies");

e.setWork("test,verify");

**break**;

**case** "Designer":

e.setTechnology("Design philosophy,Design tool used");

e.setWork("Design");

**break**;

**case** "BusinessDeveloper":

e.setTechnology("Specialized Domains");

e.setWork("Collect requirements,analyze requirements,write documents");

**break**;

**case** "TeamLeader":

e.setTechnology("Team Collaboration tools, development methods");

e.setWork("Schedule,organize,manage programmers,designers and testers");

**break**;

**case** "Architect":

e.setTechnology("Design methodology");

e.setWork("design system");

**break**;

**case** "CTO":

e.setTechnology("Technology Domains");

e.setWork("advise technologies");

**break**;

**case** "CFO":

e.setTechnology("Finance management skills");

e.setWork("Manage finance");

**break**;

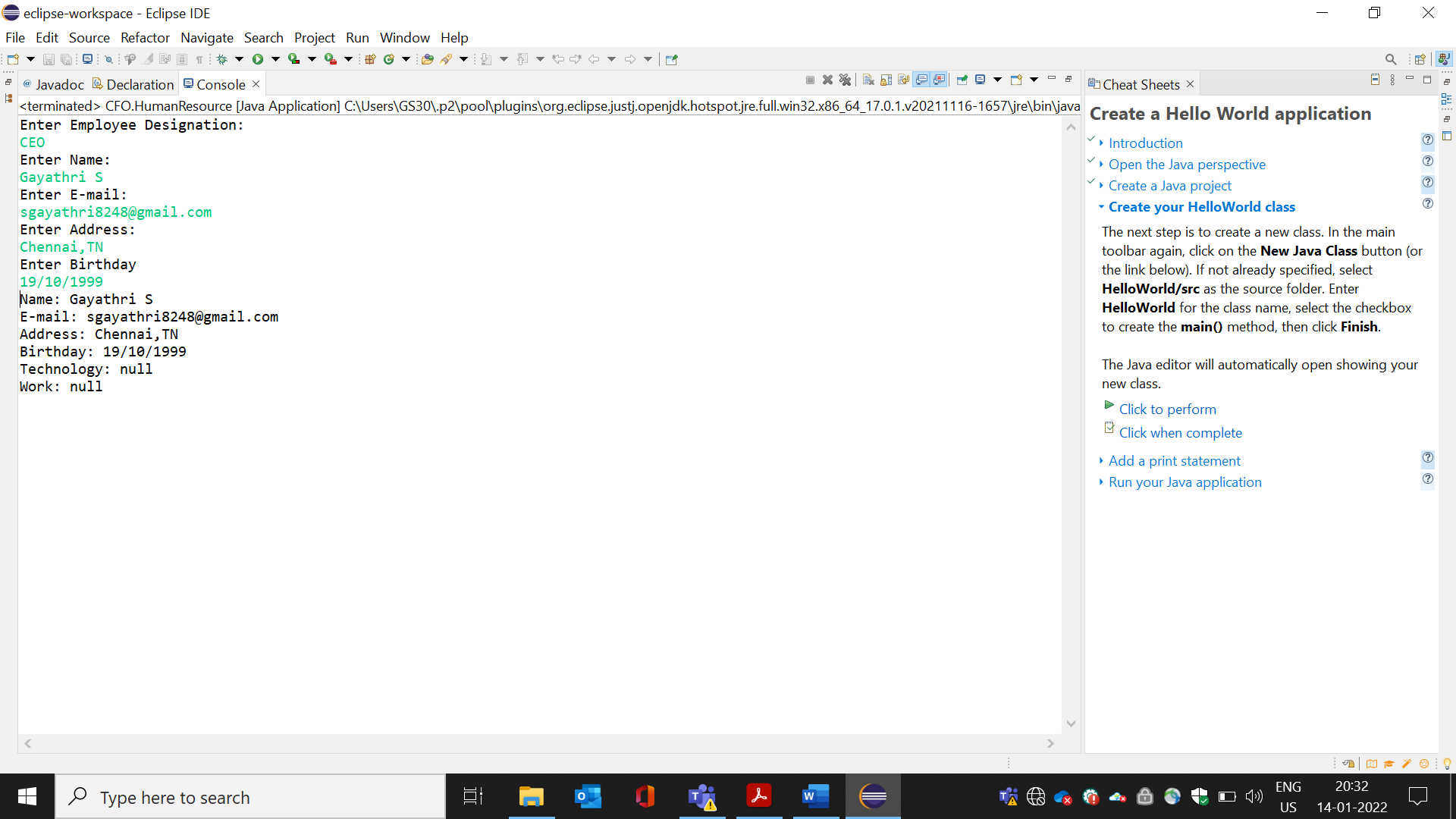
}

e.print();

}

}

**OUTPUT :**



**Project #3: Numbers to Words**

Develop a program that is able to read an arbitrary integer number from the command line, then

prints out that number in words. For example:

- **Input**: 20 **Output**: twenty

- **Input**: 86 **Output**: eighty six

- **Input**: 365 **Output**: three hundred and sixty five

The program can read numbers up to 9999. Design this program in a way that can be re-used in

other projects.

**SOURCE CODE :**

**package** number;

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

**public** **class** NumberToWords{

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

**int** n;

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

System.***out***.println("Enter the number (3-digit number)");

n=sc.nextInt();

**if**(n<1 || n>999)

{

System.***out***.println("Wrong Input,Sorry");

}

**else**

{

**int** u=n%10;

**int** tt=n/10;

**int** t=tt%10;

**int** h=n/100;

String uw[]= {"","one","two","three","four","five","six","seven","eight","nine","ten","eleven","twelve","thirteen","fourteen","fifteen","sixteen","seventeen","nineteen",};

String tw[]={"","ten","twenty","thirty","fourty","fifty","sixty","seventy","eighty","ninety"};

String hw="hundred";

**if**(n<20)

{

System.***out***.println(uw[n]);

}

**else** **if**(n<100)

{

System.***out***.println(tw[t]+" "+uw[u]);

}

**else**

{

System.***out***.println(uw[h]+hw+tw[t]+""+uw[u]);

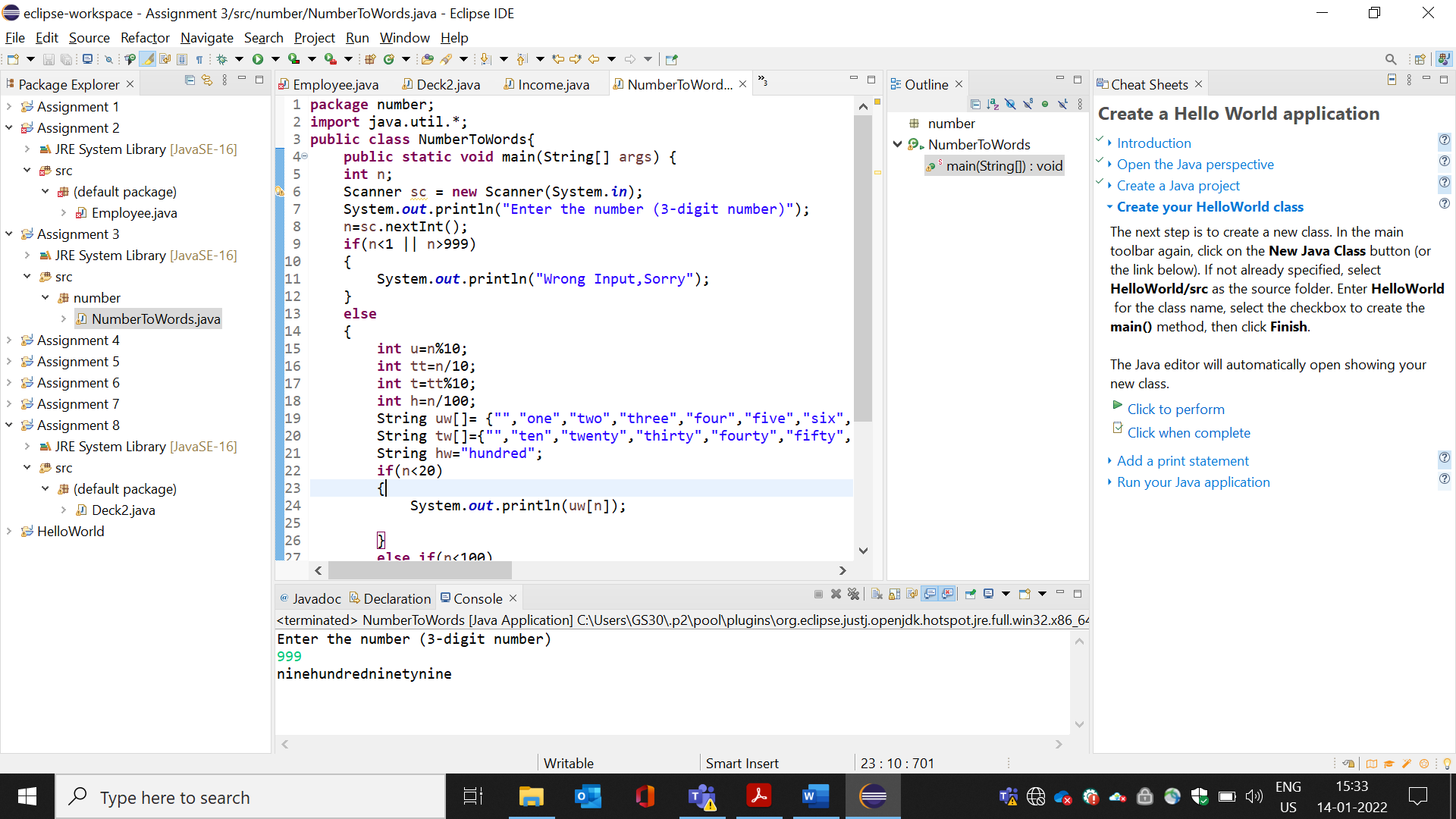
}

}

}

}

**OUTPUT :**



**Project #4: Fibonacci Sequence**

Write a program that prints the Fibonacci sequence numbers less than a given number N. For

example:

- **Input**: N = 10 **Output**: 1 1 2 3 5 8

- **Input**: N = 30 **Output**: 1 1 2 3 5 8 13 21

- **Input**: N = 60 **Output**: 1 1 2 3 5 8 13 21 34 55

**SOURCE CODE :**

**package** series;

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

**public** **class** Fibonacci{

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

{

**int** n=60,firstTerm=1,secondTerm=1;

System.***out***.println("Fibonacci Series Upto" +n+ ":");

**while**(firstTerm<=n){

System.***out***.print(firstTerm + ",");

**int** nextTerm=firstTerm + secondTerm;

firstTerm=secondTerm;

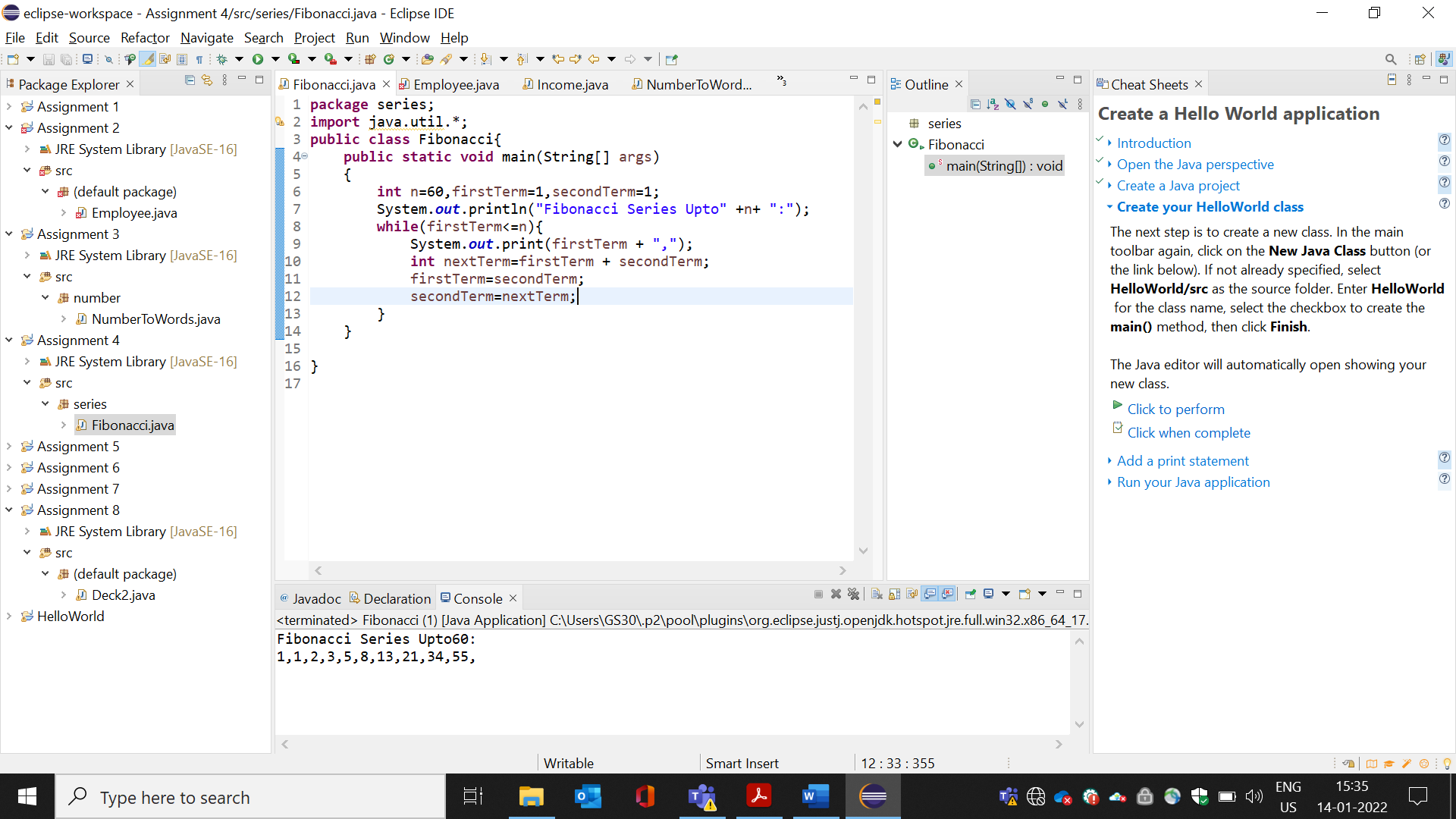
secondTerm=nextTerm;

}

}

}

**OUTPUT :**



**Project #5: Case Changer**

Write a program that can change a given input string to the following types of case:

- All upper case

- All lower case

- Capitalize (capitalize the first letter of each word)

- Sentence case (capitalize the first letter of the first word in a sentence)

- Invert: lower to upper and upper to lower

Here’s an example:

- **Input**:

Mary has a little dog. she calls him Pun

- **Output:**

Upper case:

MARY HAS A LITTLE DOG. SHE CALLS HIM PUN

Lower case:

mary has a little dog. she calls him pun

Capitalize:

Mary Has A Little Dog. She Calls Him Pun

Sentence case:

Mary has a little dog. She calls him pun

Invert:

mARY HAS A LITTLE DOG. SHE CALLS HIM pUN

**SOURCE CODE :**

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

**public** **class** Sentence {

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

{

String s1;

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

System.***out***.println("Enter a message:");

s1=sc.nextLine();

Scanner s2=**new** Scanner(s1);

String Upper\_line="";

String Lower\_line="";

String Invert\_line="";

**while**(s2.hasNext())

{

String word=s2.next();

Upper\_line +=Character.*toUpperCase*(word.charAt(0))+word.substring(1)+" ";

Lower\_line +=Character.*toLowerCase*(word.charAt(0))+word.substring(1)+" ";

Invert\_line +=Character.*toLowerCase*(word.charAt(0))+word.substring(1)+" ";

}

System.***out***.println(Upper\_line.trim());

System.***out***.println(Lower\_line.trim());

System.***out***.println(Invert\_line.trim());

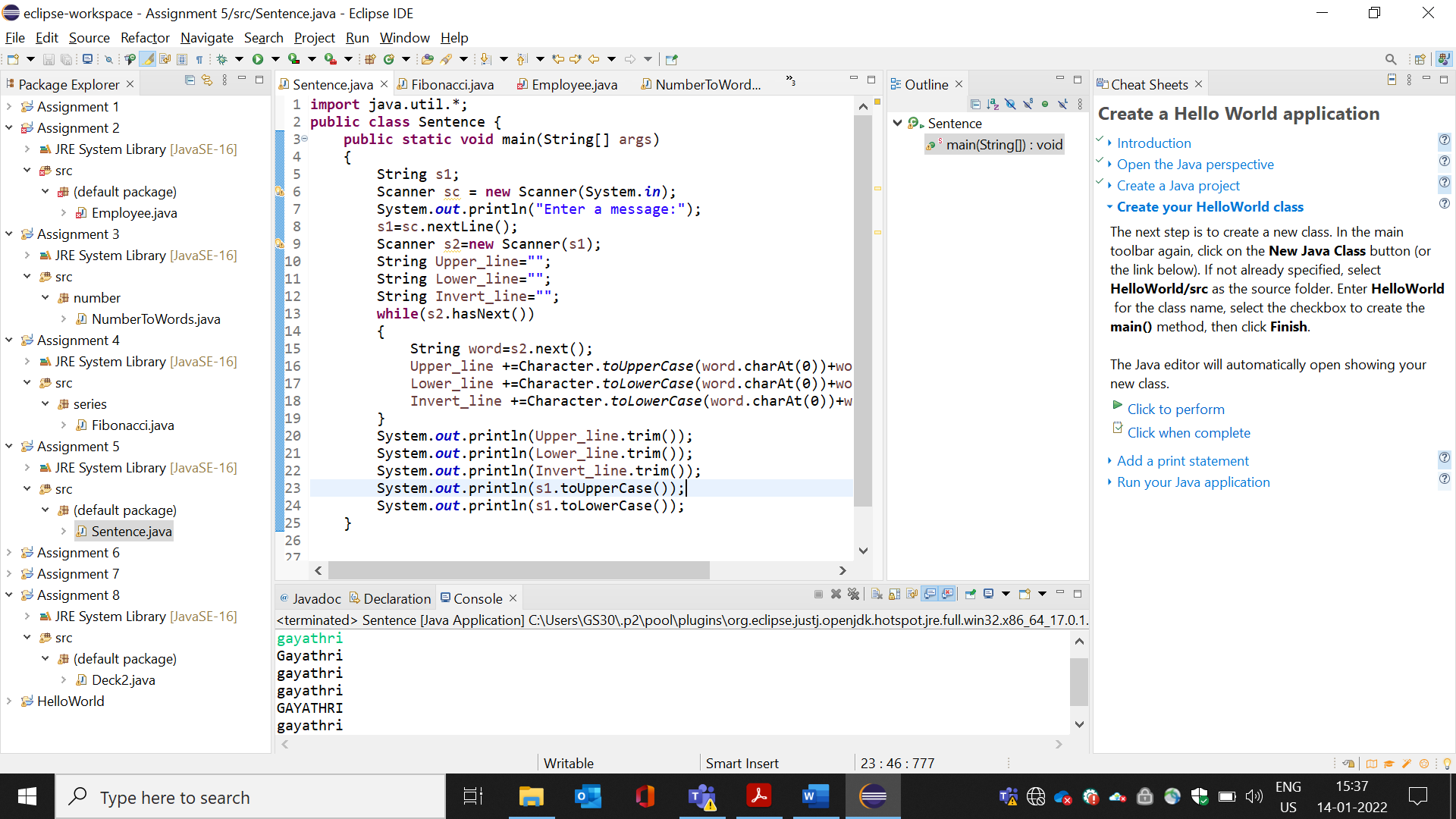
System.***out***.println(s1.toUpperCase());

System.***out***.println(s1.toLowerCase());

}

}

**OUTPUT** :



**Project #6: Factorial Number**

Write a program that calculates factorial of a given number N. It prints the output by both number

and words (reuse the stuffs developed in the project #13).

For example:

- **Input:**

N = 5

- **Output**:

5! = 120 (one hundred twenty two)

**SOURCE CODE :**

**package** number;

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

**public** **class** Factorial{

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

**int** i,fact=1;

**int** number=5;

**for**(i=1;i<=number;i++){

fact=fact\*i;

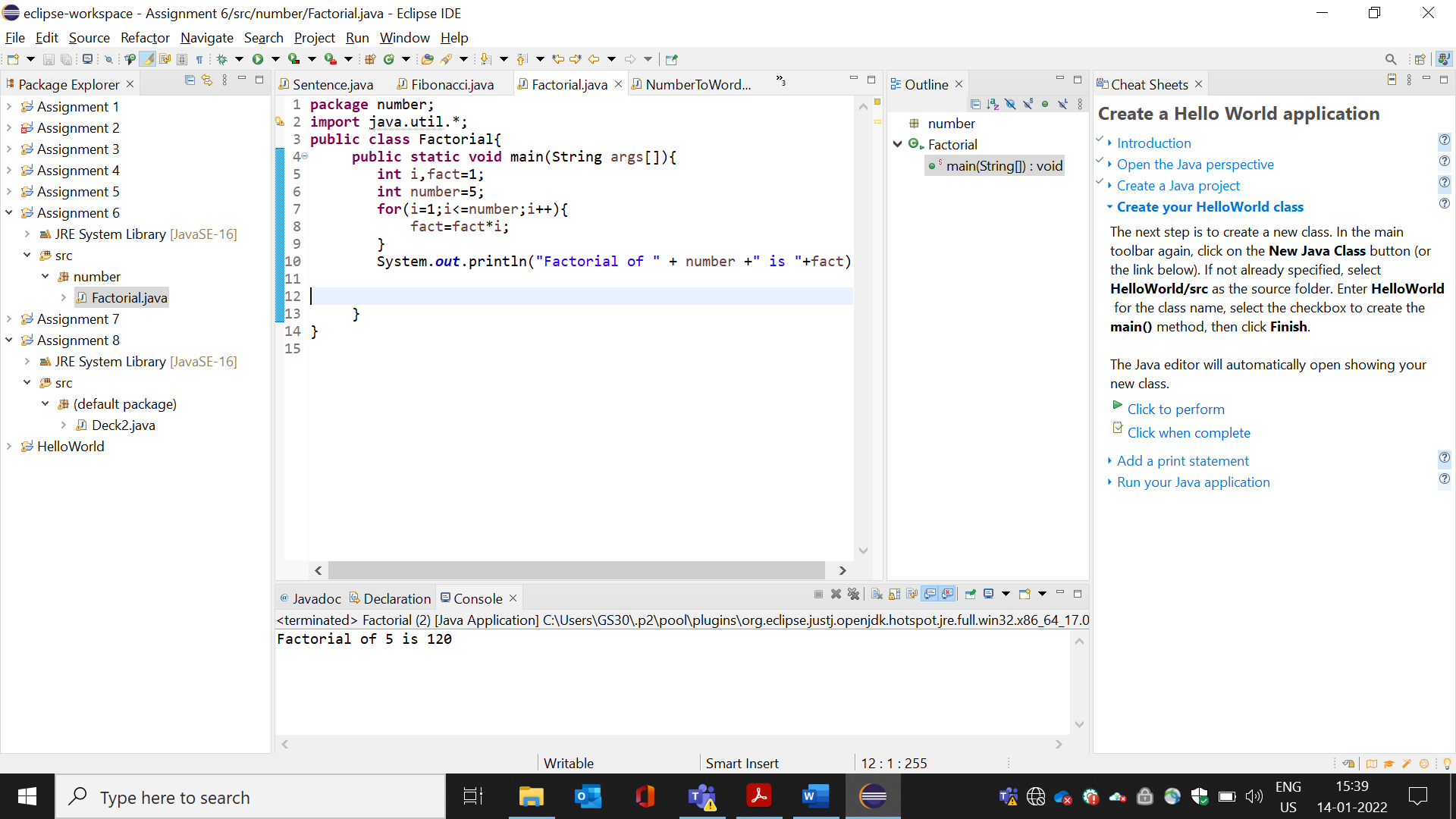
}

System.***out***.println("Factorial of " + number +" is "+fact);

}

}

**OUTPUT :**



**Project #7: Triangle Area Calculator**

Write a program that calculates area of a triangle, given its 3 points (A, B and C) in the Descartes

coordinate system.

**- Input:**

Point A (x, y): 10 20

Point B (x, y): 10 60

Point C (x, y): 80 20

**- Output**:

Area = 1400

**SOURCE CODE :**

**package** com.capg.table;

**public** **class** TriangleAreaCalculator {

**public** **static** **void** printArea(**int** x[],**int** y[]) {

**int** area=Math.*abs*(x[0] \* (y[1]-y[2]) + x[1]\*(y[2]-y[0]) + x[2]\*(y[0]-y[1]))/2;

System.***out***.println("Area of triangle ="+area);

}

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

**int** [] x\_coordinate = {10,10,80};

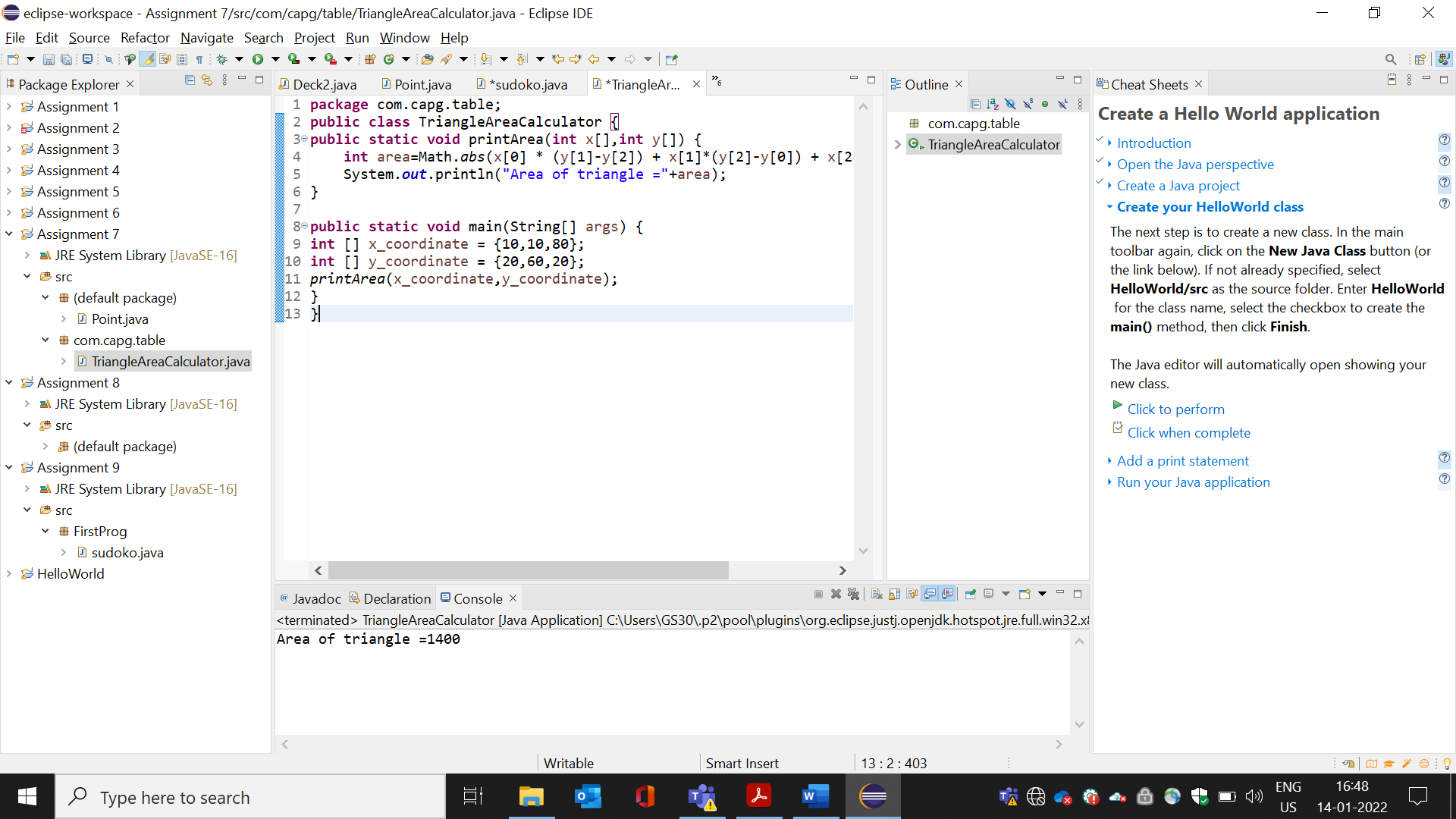
**int** [] y\_coordinate = {20,60,20};

*printArea*(x\_coordinate,y\_coordinate);

}

}

**OUTPUT :**



**Project #8: Cards Dealer**

Write a program that mimics a dealer in cards game. In each game, the dealer divides 52 cards for 4

players randomly. Suppose the following letters denote the suits:

- C: clubs - D: Diamond - H: Heart - S: Spades Here’s an

example:

Player #1: 3c 8d 2h As Jc 10h Kh 5s 6s 9d Qd 2s 7c

Player #2: 6d 6h 2d 3s 4d 4h 5c Qc Kd Jd 8c 10c Ac

Player #3: 3d 4c Jh 6c 10d 5h 7s 2c 7h Ad 8s Qh 3h

Player #4: Ah 4s 10s 5d Qs Kc Ks 9c 9h 9s Js 8h 7d

**Source code:**

**public** **class** Point {

**int** x;

**int** y;

**public** Point(**int** x,**int** y)

{

**this**.x=x;

**this**.y=y;

}

}

**class** Tri{

**public** **static** **float** area(Point A,Point B,Point C)

{

**float** ae;

ae=(A.x\*(B.y-C.y)+B.x\*(C.y-B.y)+C.x\*(A.y-B.y))/2.0f;

**return** Math.*abs*(ae);

}

**public** **static** **float** area (**int** base,**int** height)

{

**return**(base\*height)/2.0f;

}

}

**class** Area

{

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

{

Point A=**new** Point(10,20);

Point B=**new** Point(10,60);

Point C=**new** Point(80,20);

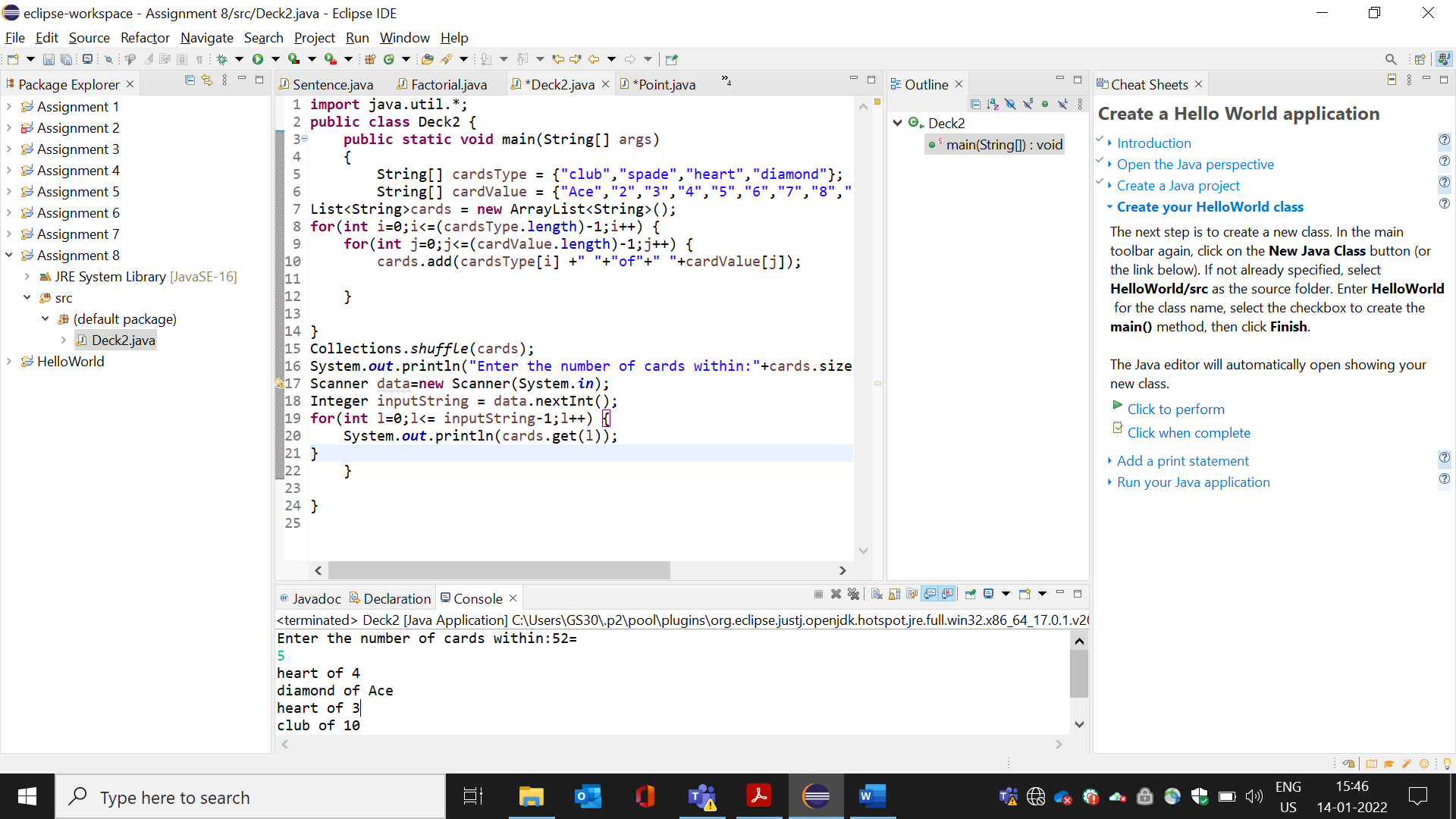
System.***out***.println("Area of Triangle formula 1:"+Tri.*area*(A,B,C));

System.***out***.println("Area of Triangle formula 1:"+Tri.*area*(3,5));

}

}

**OUTPUT :**



**Project #9: Sudoku Matrix**

Write a program that randomly prints a Sudoku board (9x9 matrix) which is divided into 3 submatrices

(3x3 each). Each sub matrix contains numbers ranging from 1 to 9.

Fill numbers into the whole board in ways so that there is no duplicate numbers on each row, each

column and each 3x3 block. Here’s an example:

SOURCE CODE :

**package** FirstProg;

**public** **class** sudoko {

**private** **static** **final** **int** ***GRID\_SIZE*** = 9;

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

**int** [][] board = {

{7,0,2,0,5,0,6,0,0},

{0,0,0,0,0,3,0,0,0},

{1,0,0,0,0,9,5,0,0},

{8,0,0,0,0,0,0,9,0},

{0,4,3,0,0,0,7,5,0},

{0,9,0,0,0,0,0,0,8},

{0,0,9,7,0,0,0,0,5},

{0,0,0,2,0,0,0,0,0},

{0,0,7,0,4,0,2,0,3}

};

*printBoard*(board);

**if**(*solveBoard*(board)) {

System.***out***.println("Solved successfully!");

}

**else** {

System.***out***.println("Unsolvable board :");

}

*printBoard*(board);

}

**private** **static** **void** printBoard(**int** [][] board) {

**for** (**int** row=0;row<***GRID\_SIZE***;row++) {

**if**(row % 3==0 && row!=0) {

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

}

**for**(**int** column =0; column<***GRID\_SIZE***;column++) {

**if**(column %3==0 && column !=0) {

System.***out***.print("|");

}

System.***out***.print(board[row][column]);

}

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

}

}

**private** **static** **boolean** isNumberInRow(**int** [][] board,**int** number,**int** row) {

**for** (**int** i=0;i<***GRID\_SIZE***;i++) {

**if**(board[row][i]==number) {

**return** **true**;

}

}

**return** **false**;

}

**private** **static** **boolean** isNumberInColumn(**int** [][] board,**int** number,**int** column) {

**for**(**int** i=0;i<***GRID\_SIZE***;i++) {

**if**(board[i][column]==number) {

**return** **true**;

}

}

**return** **false**;

}

**private** **static** **boolean** isNumberInBox(**int**[][] board,**int** number,**int** row,**int** column)

{

**int** localBoxRow=row-row%3;

**int** localBoxColumn=column-column%3;

**for**(**int** i =localBoxRow;i<localBoxRow+3;i++) {

**for**(**int** j=localBoxColumn;j<localBoxColumn+3;j++) {

**if**(board[i][j]==number) {

**return** **true**;

}

}

}

**return** **false**;

}

**private** **static** **boolean** isValidPlacement(**int** [][] board,**int** number,**int** row,**int** column) {

**return** !*isNumberInRow*(board,number,row)&&

!*isNumberInColumn*(board,number,column)&&

!*isNumberInBox*(board,number,row,column);

}

**private** **static** **boolean** solveBoard(**int** [][] board) {

**for**(**int** row=0;row<***GRID\_SIZE***;row++) {

**for**(**int** column=0;column<***GRID\_SIZE***;column++) {

**if**(board[row][column]==0) {

**for**(**int** numberToTry = 1; numberToTry<=***GRID\_SIZE***;numberToTry++) {

**if**(*isValidPlacement*(board,numberToTry,row,column)) {

board[row][column]=numberToTry;

**if**(*solveBoard*(board)) {

**return** **true**;

}

**else** {

board[row][column]=0;

}

}

}

**return** **false**;

}

}

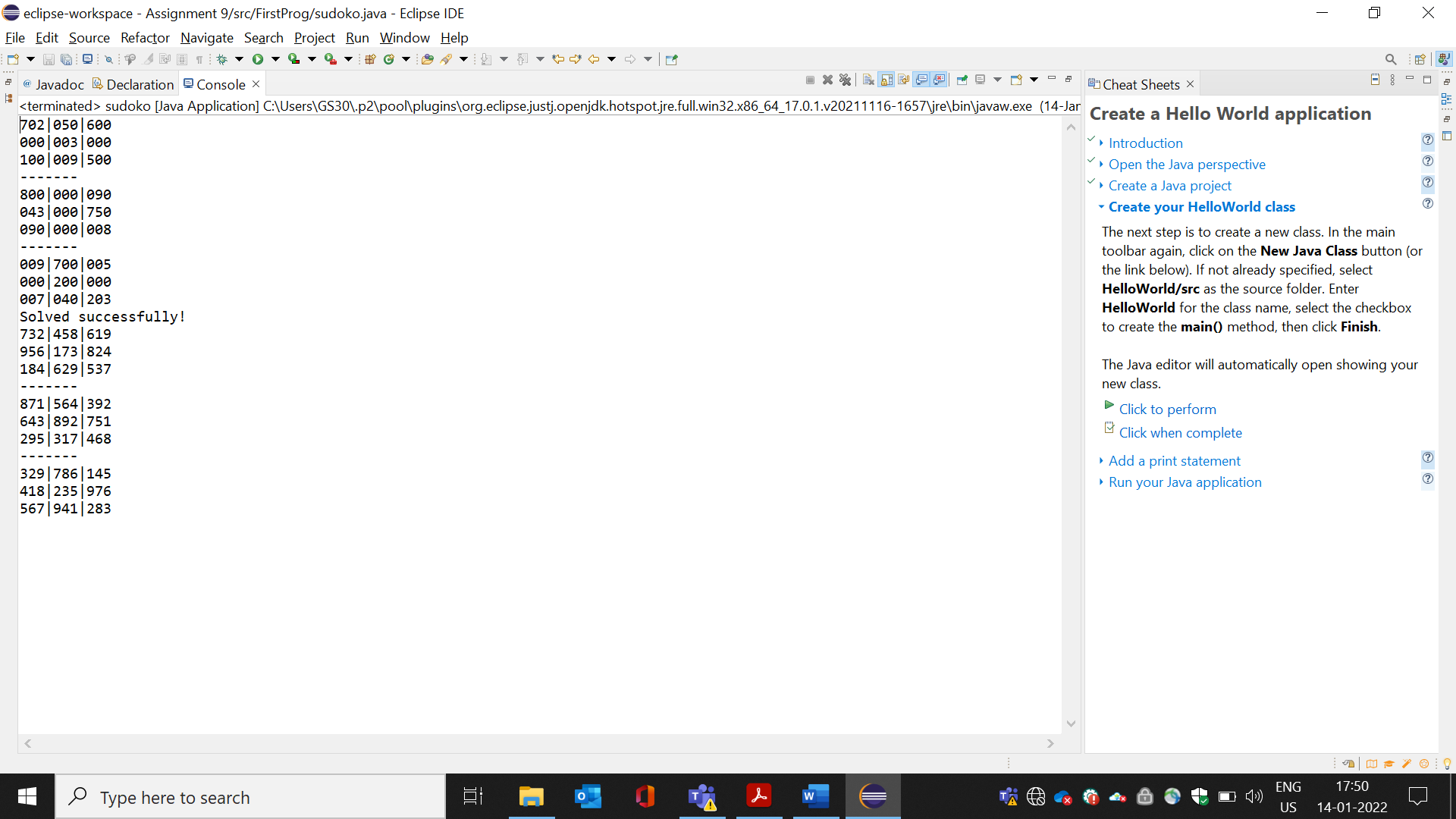
}

**return** **true**;

}

}

OUTPUT :



**Project #10: Permutation**

Write a program that prints permutations of any given set of numbers. For example:

**Input**:

Total number N = 3

Number #1: 2

Number #2: 8

Number #3: 9

**Output**:

The permutations of (2, 8, 9) are: 6

(2, 8, 9) (2, 9, 8) (8, 2, 9)

(8, 9, 2) (9, 2, 8) (9, 8, 2)

SOURCE CODE :

**package** com.capg.table;

**public** **class** Permutation {

**public** **static** **void** printArray(**int** [] a)

{

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

**for**(**int** i=0;i<a.length;i++) {

System.***out***.print(a[i]+" ");

}

}

**public** **static** **void** swap(**int**[] a, **int** i, **int** j)

{

**int** temp = a[i];

a[i] = a[j];

a[j] = temp;

}

**public** **static** **void** printPermutation(**int**[] a, **int** cid) {

**if**(cid == a.length-1)

{

{

*printArray*(a);

**return**;

}

}

**for**(**int** i=cid;i<a.length;i++) {

*swap*(a,i,cid);

*printPermutation*(a,cid+1);

*swap*(a,i,cid);

}

}

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

**int**[] a= {2,8,9};

*printPermutation*(a,0);

}

}

OUTPUT :

