FINAL VARIABLE

1. IT CANNOT BE REINITIALIZED WITH ANOTHER VALUE
2. THE FINAL METHOD CANNOT BE OVERRIDDEN
3. THE FINAL CLASS CANNOT BE OVERRIDDEN

class Main {

public static void main(String[] args) {

final int AGE = 32;

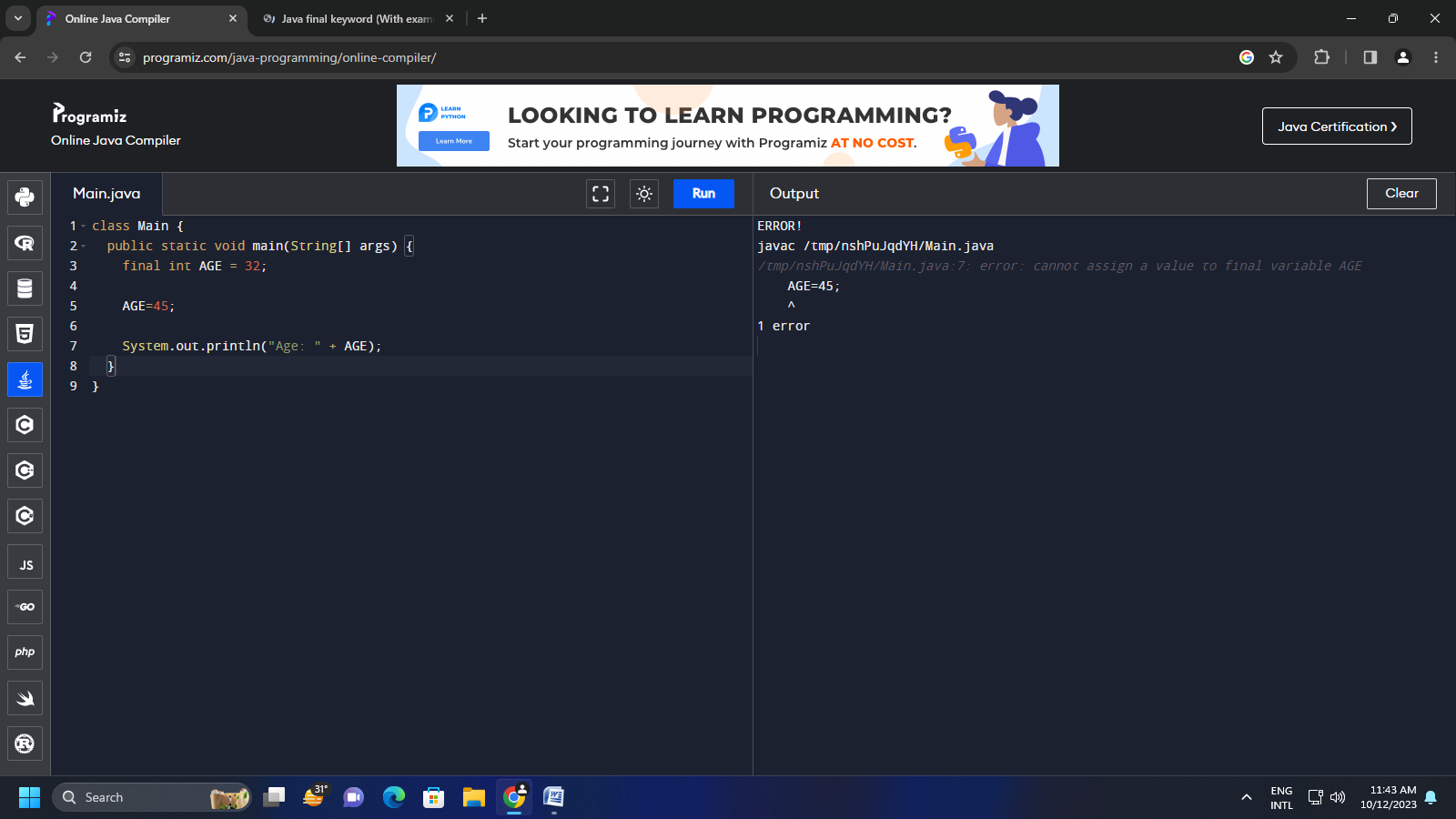
AGE=45;

System.out.println("Age: " + AGE);

}

}

Error at AGE=45;



class FinalDemo {

public final void display() {

System.out.println("This is a final method.");

}

}

class Main extends FinalDemo {

public final void display() {

System.out.println("The final method is overridden.");

}

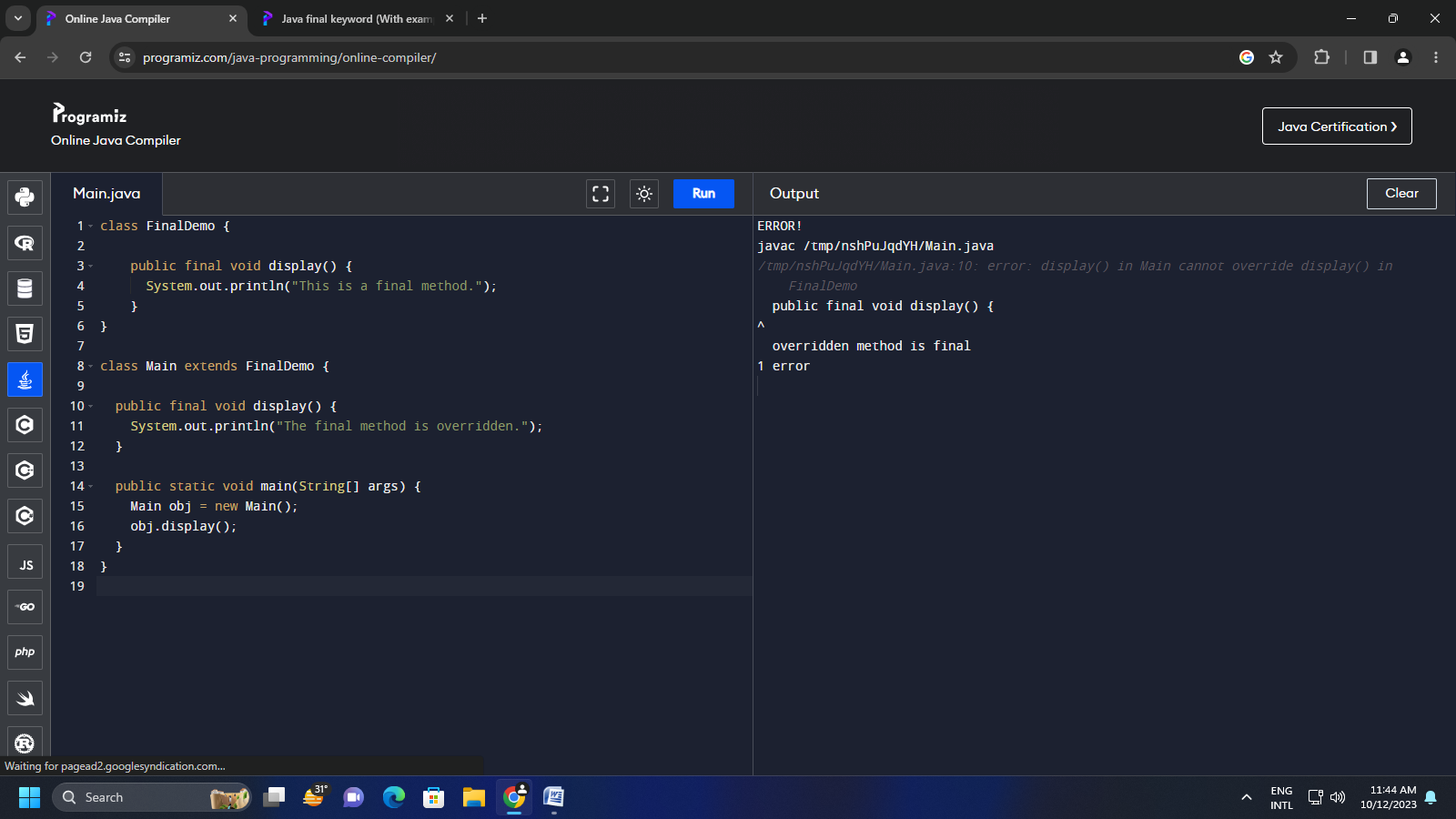
public static void main(String[] args) {

Main obj = new Main();

obj.display();

}

}



final class Finalclass {

public final void display() {

System.out.println("This is a final method.");

}

}

class Main extends Finalclass {

public final void display() {

System.out.println("The final method is overridden.");

}

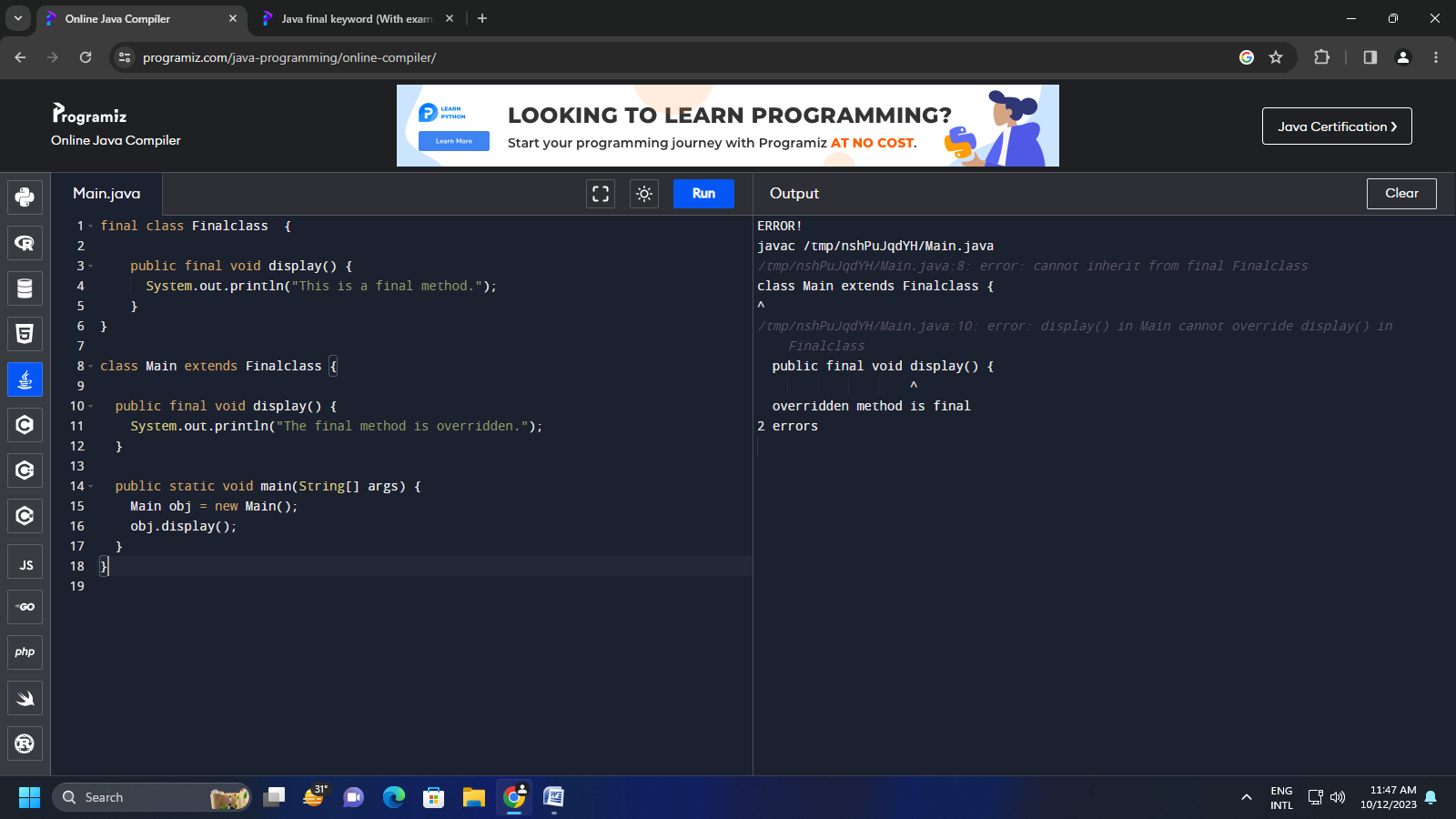
public static void main(String[] args) {

Main obj = new Main();

obj.display();

}

}



1)FINAL CLASS CANT BE INHERITED

CONSTRUCTOR CHAINING

public class Constructorchain{

Constructorchain()

{

this ("hello");

System.out.println("default constructor called");

}

Constructorchain(String str)

{

System.out.println("parameterized constructor called");

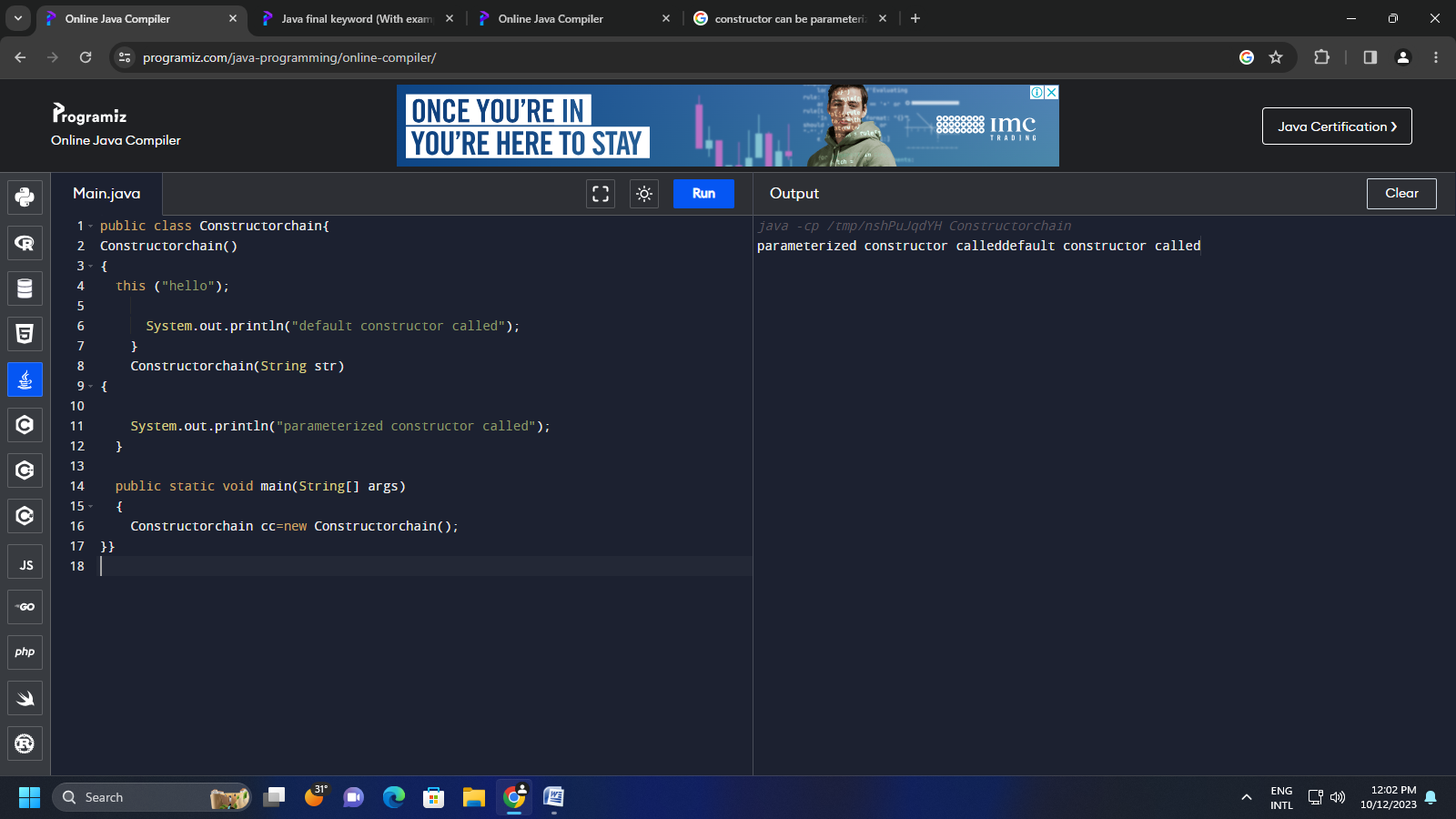
}

public static void main(String[] args)

{

Constructorchain cc=new Constructorchain();

}}



INHERITANCE

public class automobile{

public void start()

{

System.out.println("engine starts runninng");

}

public void stop()

{

System.out.println("engine stops running");

}

class car extends automobile

{

System.out.println("4 tyres");

}

public void fuel()

{

System.out.println("petrol");

}

public void fuel(String fuel)

{

System.out.println("also run on diesel");

}

public void seating ()

{

System.out.println("4 seater");

}

class bike extends automobile

{

public void tyre()

{

System.out.println("2 tyre");

}

class main

{

public static void main(String[] args) {

car c = new car()

bike b= new bike()

c.fuel()

b.seat()

}}INTERFACE

interface solution

{

public void Hello();

public void Welcome();

}

public class A implements solution

{

public void Hello()

{

System.out.println("hello world");

}

public void Welcome()

{

System.out.println("welcome sir");

}

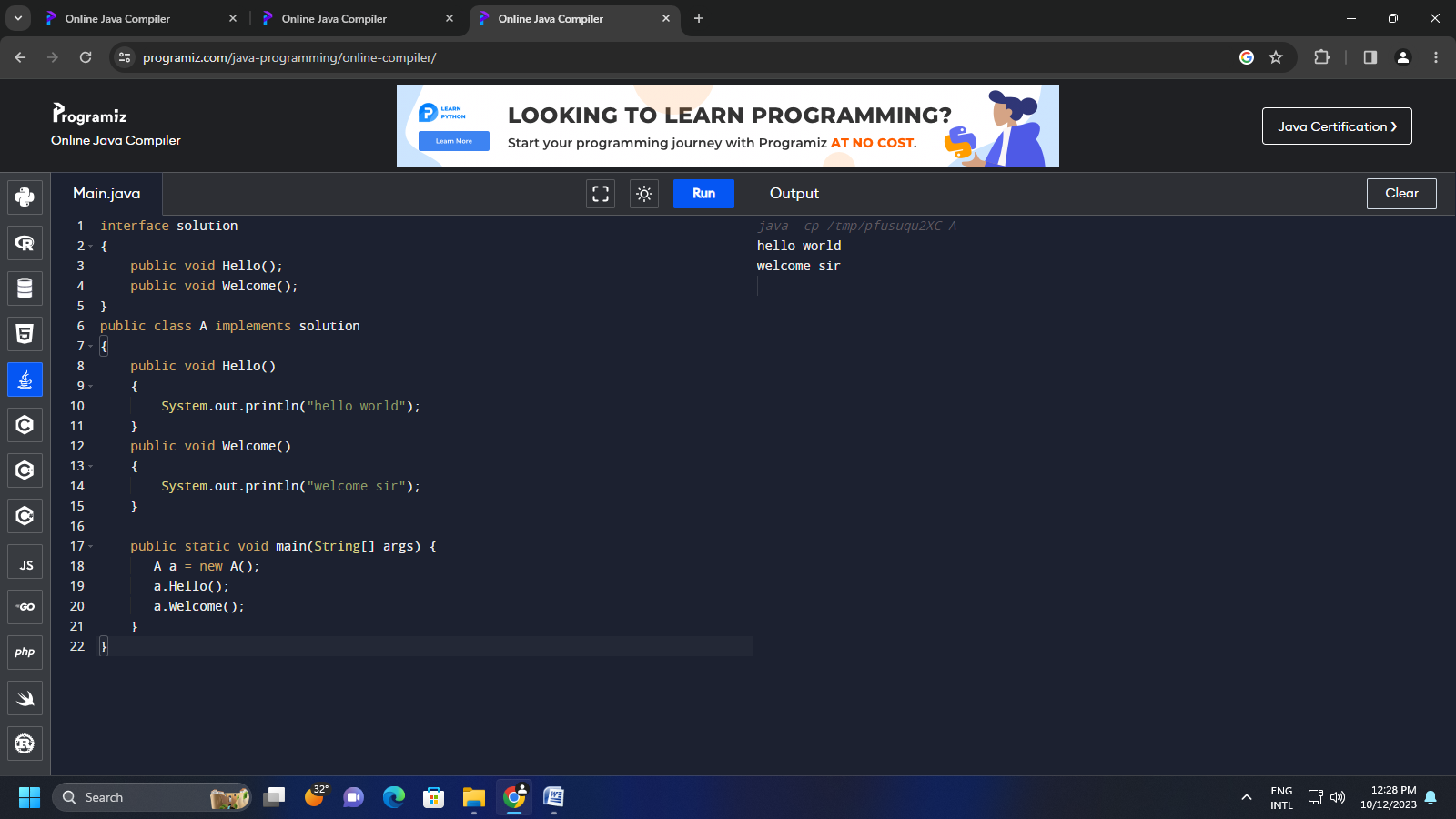
public static void main(String[] args) {

A a = new A();

a.Hello();

a.Welcome();

}}



Make a calculator using interface to implement sum ,add ,subtract, multiply .(TASK)

public interface Calculator {

public int add(int a, int b);

public int subtract(int a, int b);

public int multiply(int a, int b);

}

//user implements calculator

public class Calculator implements Calculator {

public int a

public int b

public int add(int a, int b)

System.out.println("Calculator - add " + a + b);

return a + b;

}

public int subtract(int a, int b) {

System.out.println("Calculator - subtract " + a + b);

return a - b;

}

public int multiply(int a, int b) {

System.out.println("Calculator - multiply " + a + b);

return a \* b;

}