CS 673 Software Engineering

Spring 2023

Assignment 1 – Coding

**Due:** February 1, 2023, before class.

Do not email homework, please submit on Blackboard. You should also take screenshots or recording of your assignment and submit. Post the code on GitHub and submit the link and the videos in a zip file. **This assignment must be done and submitted individually.**

In this course each student will be using the concepts and techniques learned during the semester to build a software product. Everyone must contribute code that they developed. While there is flexibility in the language your team chooses (eg. Java, Python, Kotlin) you must use good object-oriented design and coding practices. The example below demonstrates the fundamental concept that you don’t write code for the computer – you write it for other humans (maybe you!). Choose names that explain things, add comments where these names don’t suffice. Never comment on what the code is doing, only write comments that explain why.

This is a bad example:

// This function checks whether a number is even

def f(x):

// compute x modulo 2 and check whether it is zero

if modulo(x,2) == 0:

// the number is even

return True

else:

// the number is odd

return False

The exact same idea is much easier to understand if you write it like this:

def is\_divisible(number, divisor):

return modulo(number, divisor) == 0

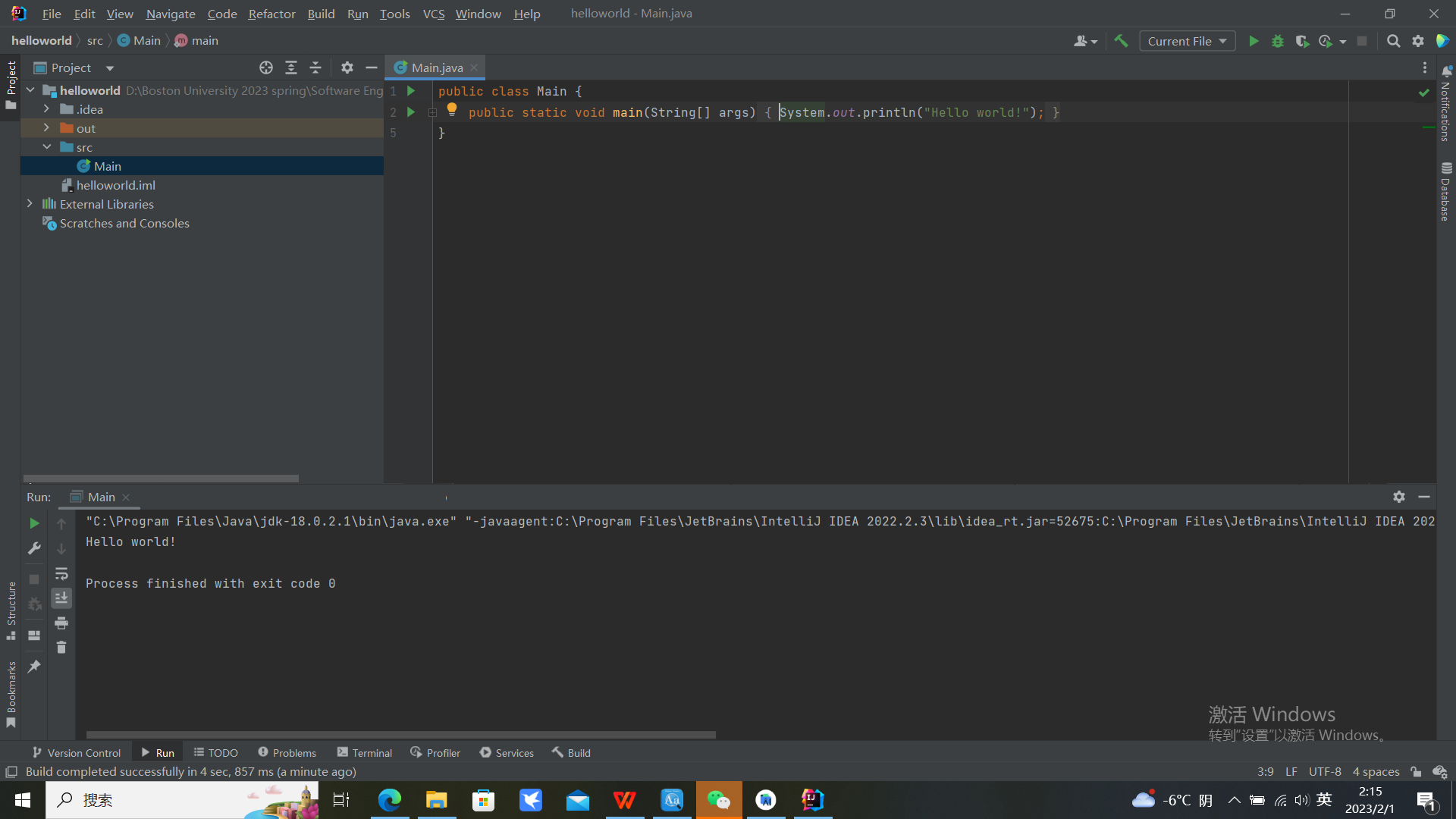
def is\_even(number):

return is\_divisible(number, 2)

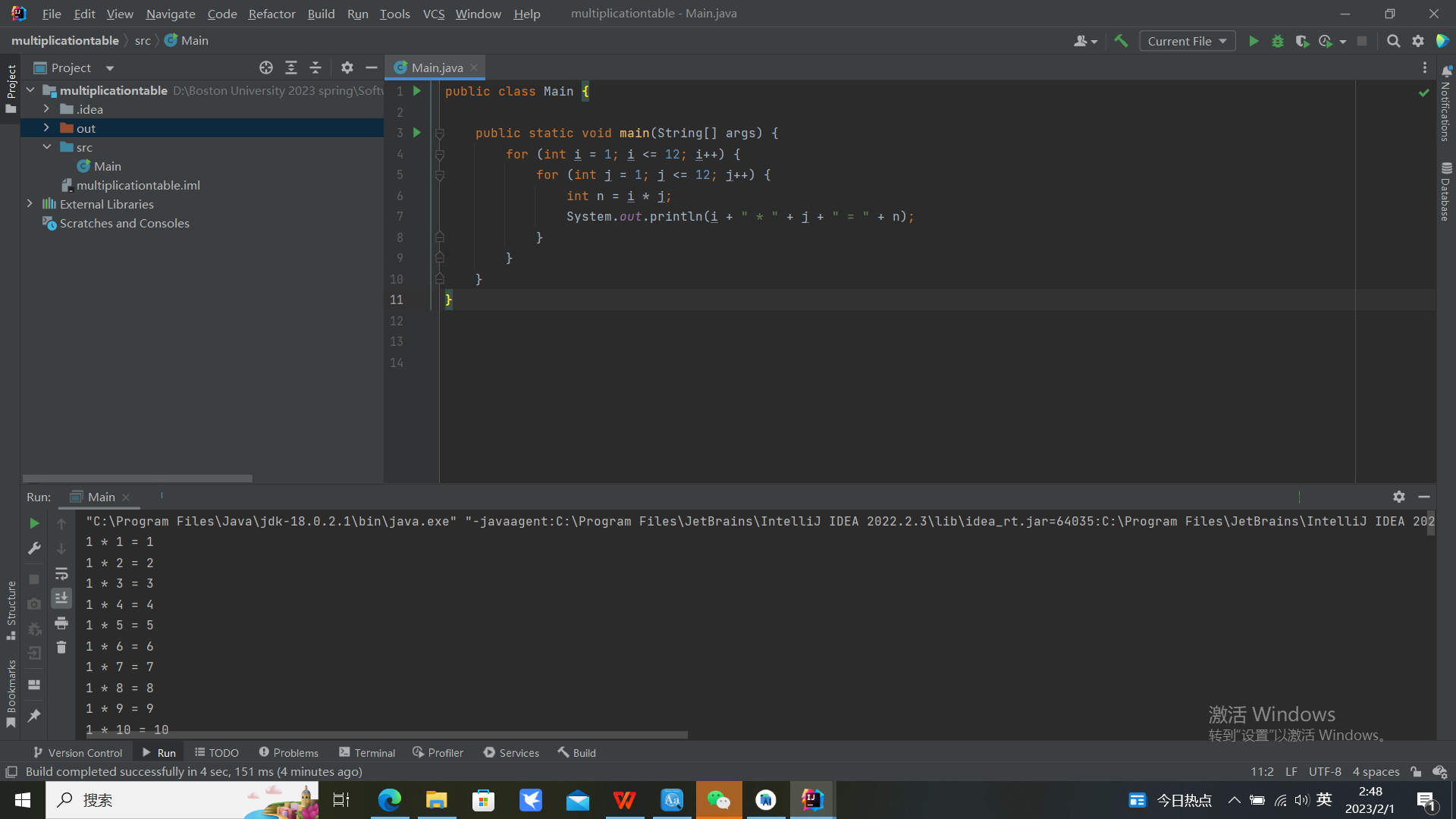
Better naming and a better task breakdown make the comments obsolete.

**Assignments:**

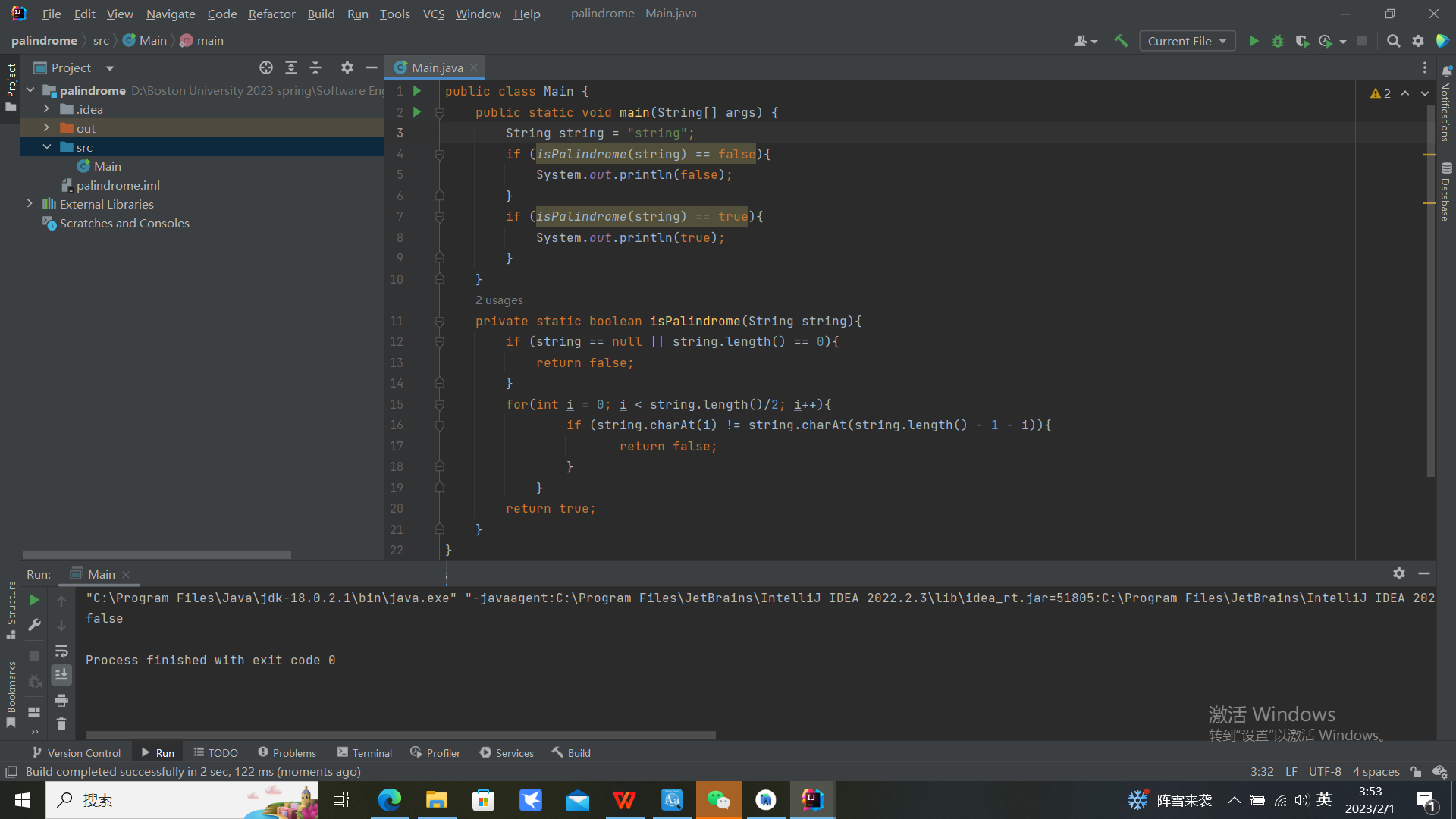
1. Read all of chapters 1 and 2 before the next class.
2. Write a program that prints ‘Hello World’ to the screen.



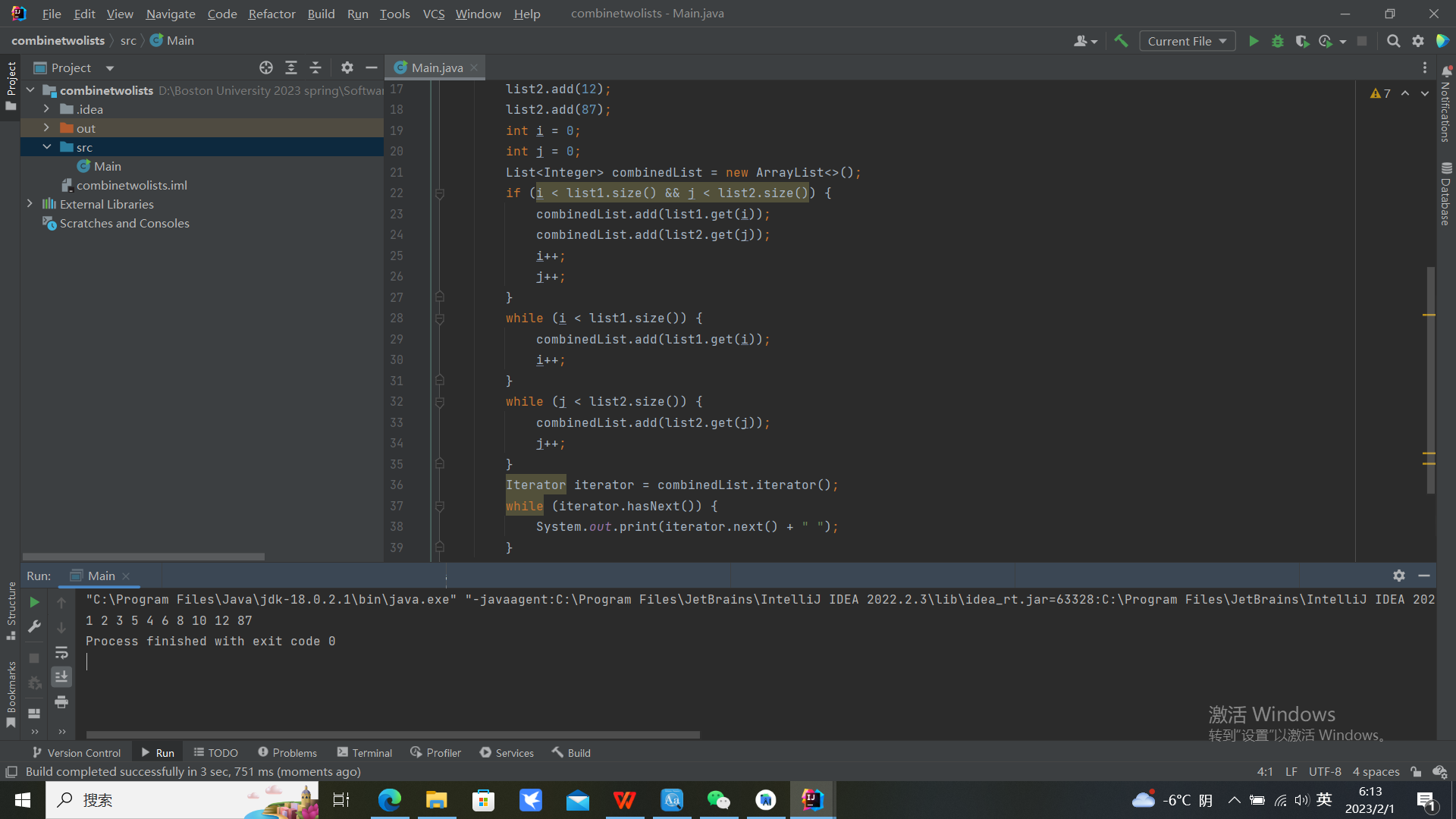
1. Write a program that prints a multiplication table for numbers up to 12.



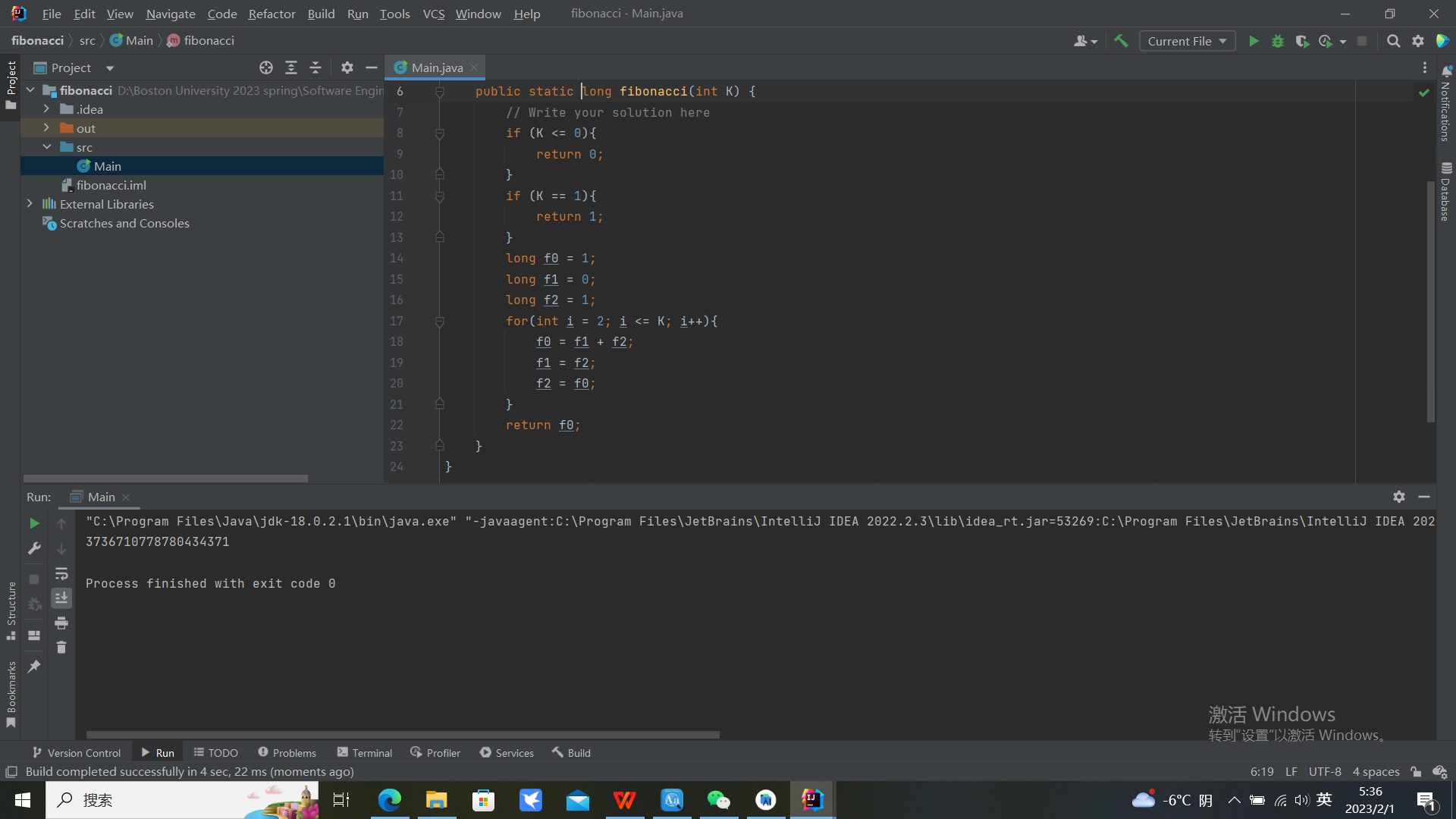
1. Write a function that tests whether a string is a palindrome and test it in your program.



1. Write a function that combines two lists by alternatingly taking elements, e.g. [a,b,c], [1,2,3] → [a,1,b,2,c,3] and test it in your program.



1. Write a function that computes the list of the first 100 Fibonacci numbers. The first two Fibonacci numbers are 1 and 1. The n+1-st Fibonacci number can be computed by adding the n-th and the n-1-th Fibonacci number. The first few are therefore 1, 1, 1+1=2, 1+2=3, 2+3=5, 3+5=8. Test it in your program.



1. Write a function that determines if a given year is a leap year. Test it in your program.

