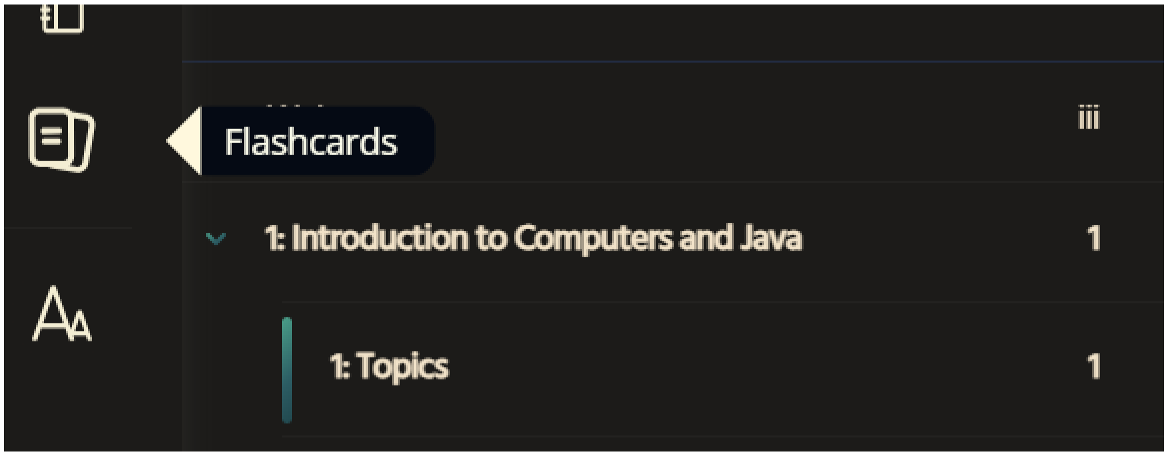
Class Notes 09/19

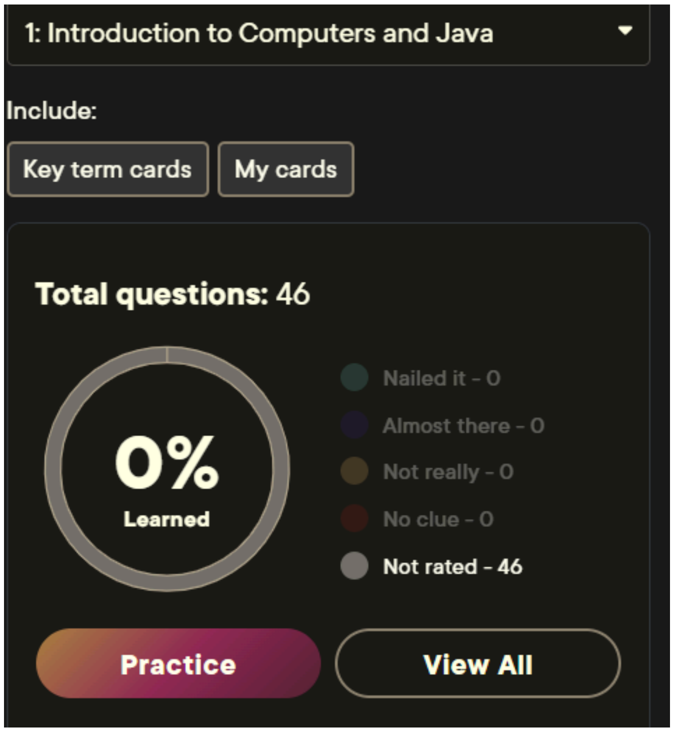
Went over homework
– problems 1 and 3 should be handwritten and can be checked on compiler

Covered some information regarding book
– how to get to flashcards

Flashcards for Ch 1



1: Introduction to Computers and Java	1
1: Topics	1



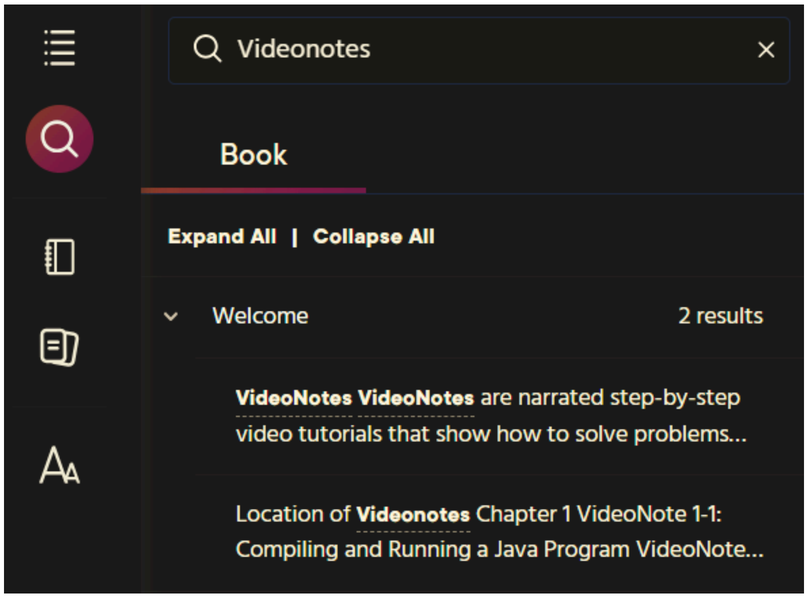
Total questions: 46

0% Learned

- Nailed it - 0
- Almost there - 0
- Not really - 0
- No clue - 0
- Not rated - 46

How to get to Video Notes

- **Go to search**
- **Enter "Video Notes"**
- **Click on Location of VideoNotes**



Videonotes

Book

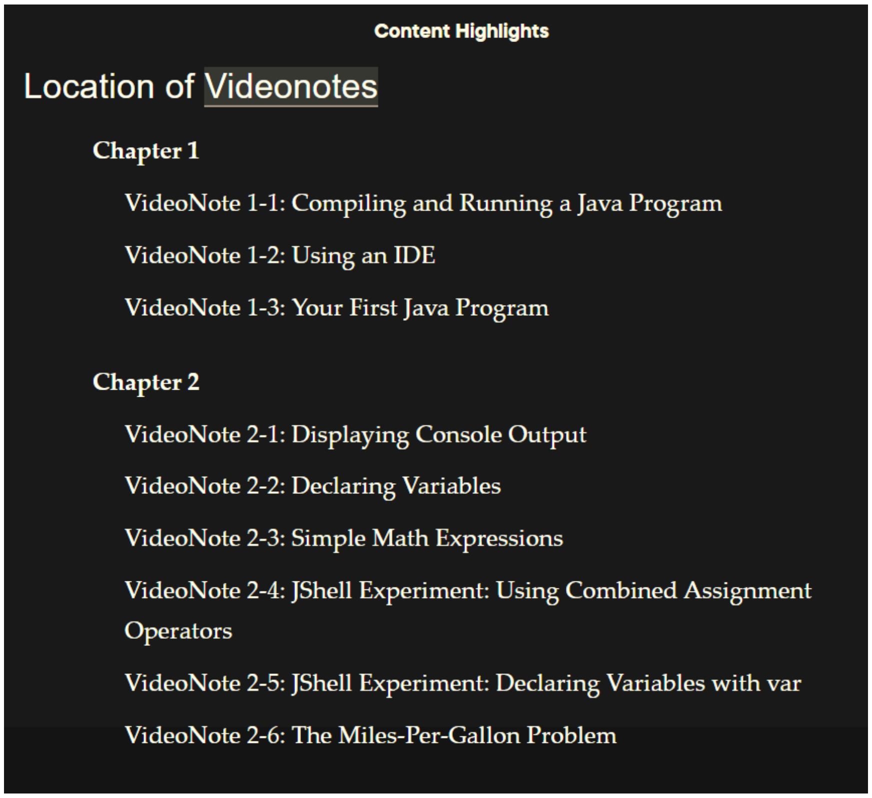
Expand All | Collapse All

Welcome 2 results

VideoNotes VideoNotes are narrated step-by-step video tutorials that show how to solve problems...

Location of Videonotes Chapter 1 VideoNote 1-1: Compiling and Running a Java Program VideoNote...

This goes down by each chapter



Content Highlights

Location of Videonotes

Chapter 1

VideoNote 1-1: Compiling and Running a Java Program

VideoNote 1-2: Using an IDE

VideoNote 1-3: Your First Java Program

Chapter 2

VideoNote 2-1: Displaying Console Output

VideoNote 2-2: Declaring Variables

VideoNote 2-3: Simple Math Expressions

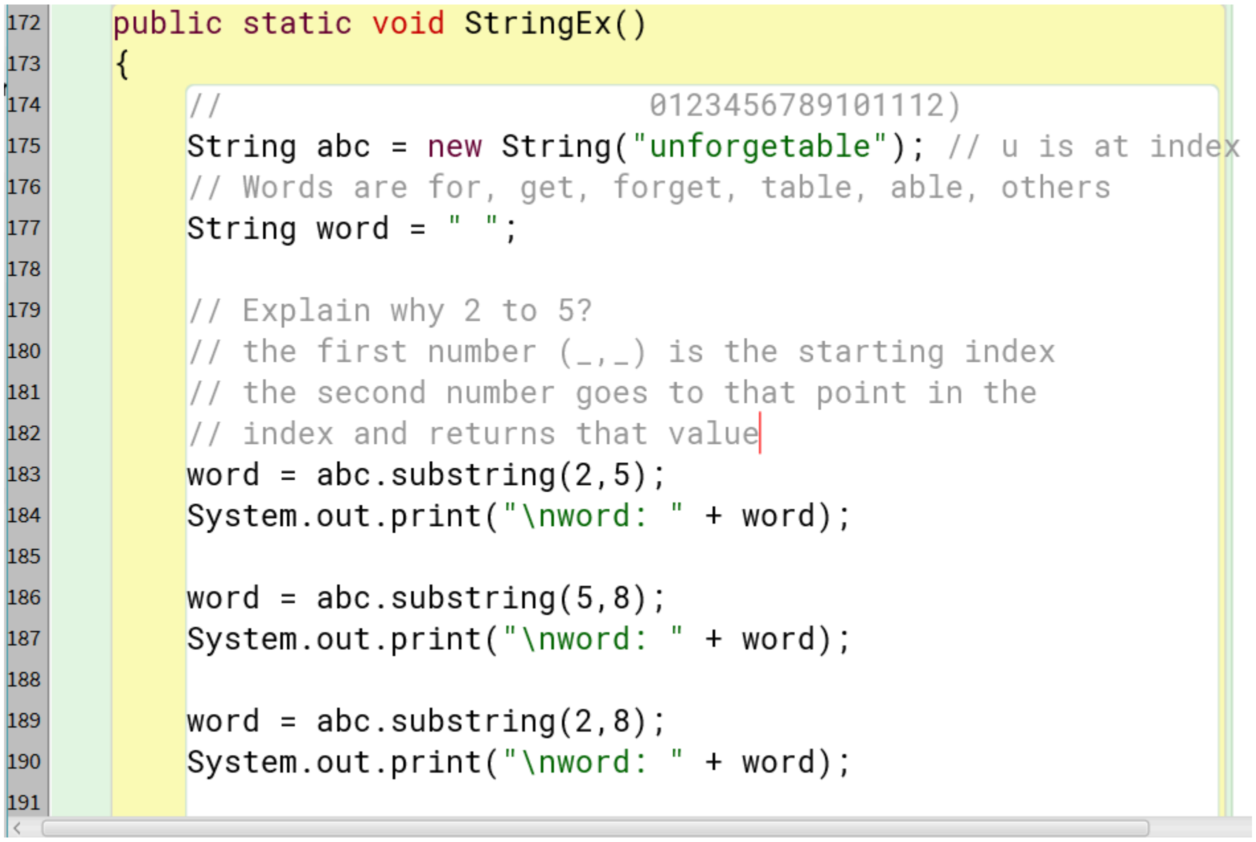
VideoNote 2-4: JShell Experiment: Using Combined Assignment Operators

VideoNote 2-5: JShell Experiment: Declaring Variables with var

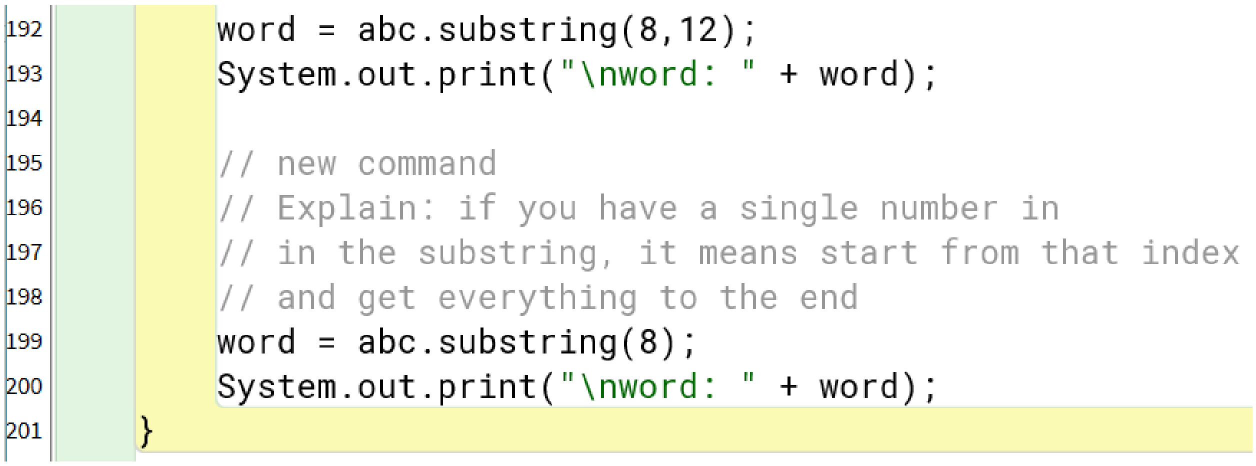
VideoNote 2-6: The Miles-Per-Gallon Problem

We continued Sect3_4YourNames

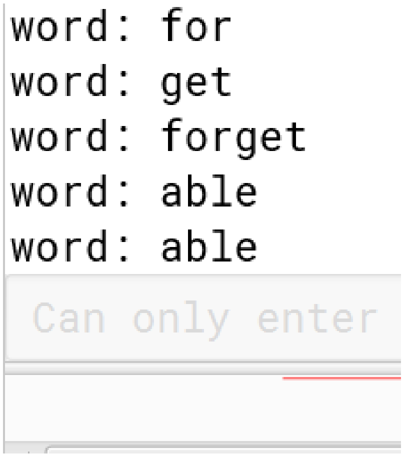
What subwords do you see in the String “unforgetable”?
For, get, forget, table, able, ect.



```java
    public static void StringEx()
    {
        //                       0123456789101112)
        String abc = new String("unforgetable"); // u is at index
        // Words are for, get, forget, table, able, others
        String word = " ";

        // Explain why 2 to 5?
        // the first number (_,_) is the starting index
        // the second number goes to that point in the
        // index and returns that value
        word = abc.substring(2,5);
        System.out.print("\nword: " + word);

        word = abc.substring(5,8);
        System.out.print("\nword: " + word);

        word = abc.substring(2,8);
        System.out.print("\nword: " + word);

```

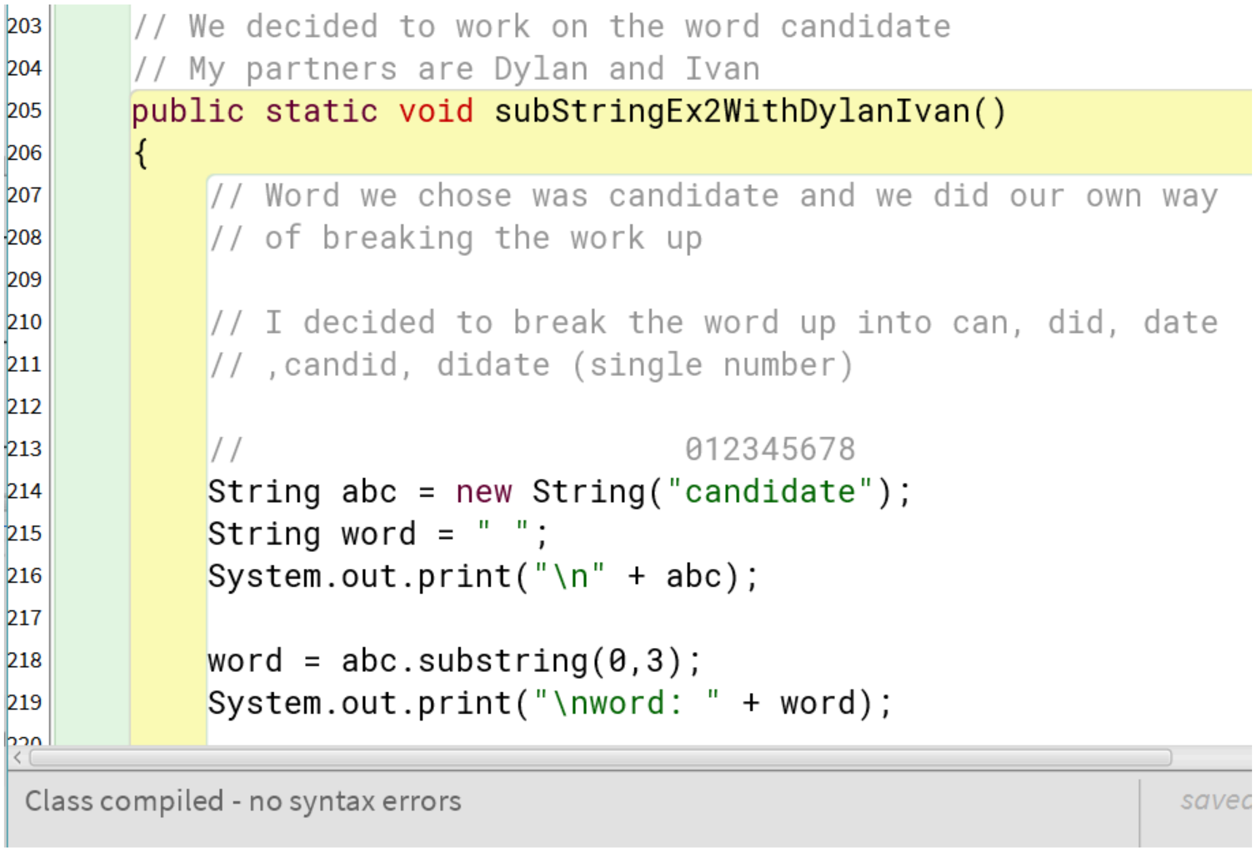


```java
        word = abc.substring(8,12);
        System.out.print("\nword: " + word);

        // new command
        // Explain: if you have a single number in
        // in the substring, it means start from that index
        // and get everything to the end
        word = abc.substring(8);
        System.out.print("\nword: " + word);
    }
```



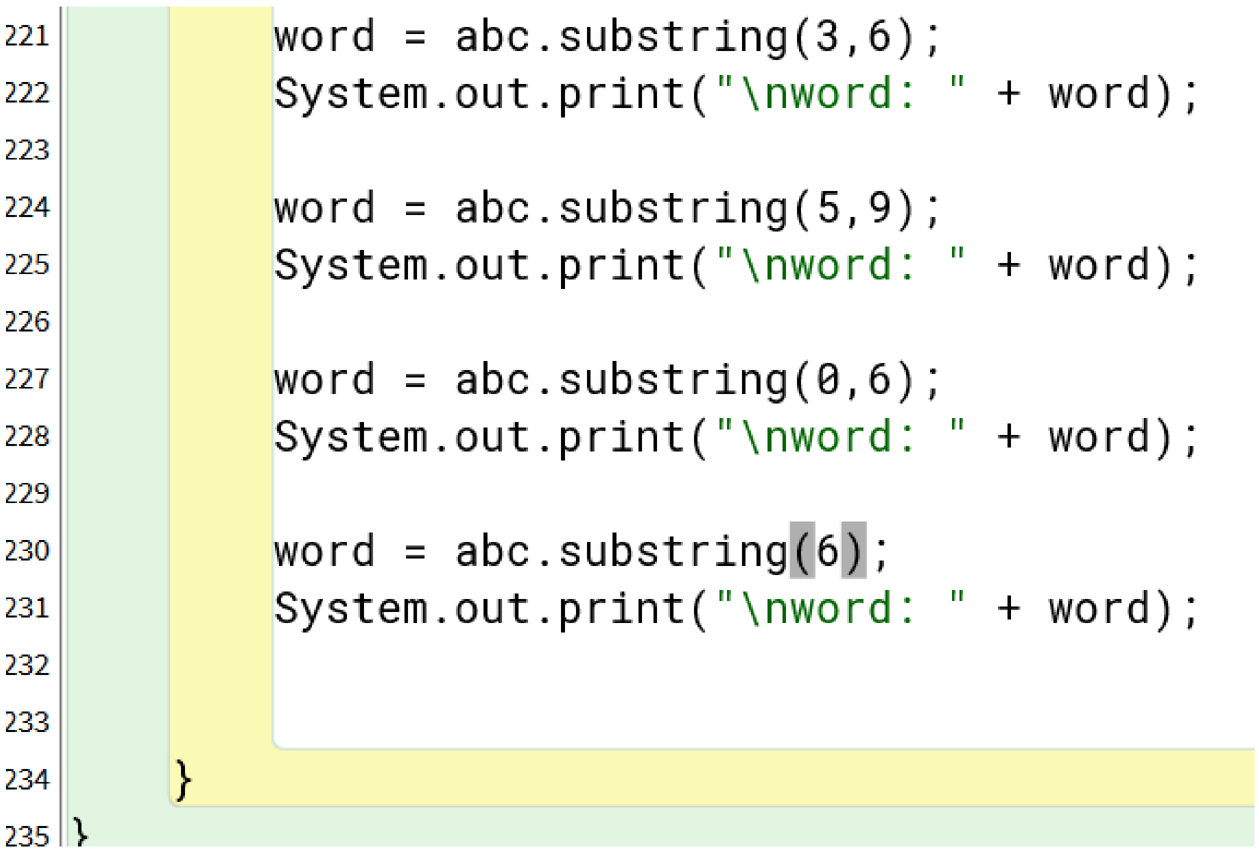
```
word: for
word: get
word: forget
word: able
word: able
```

This was our example



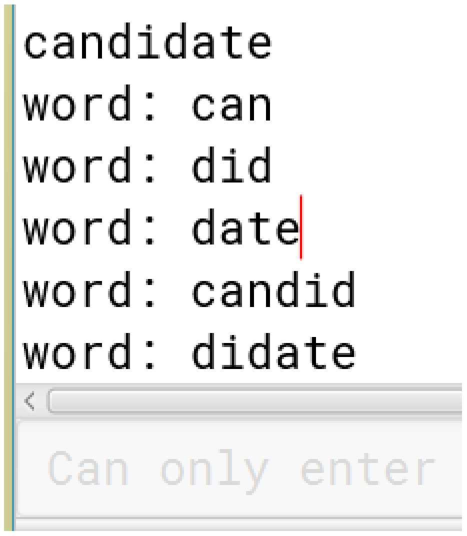
```
// We decided to work on the word candidate
// My partners are Dylan and Ivan
public static void subStringEx2WithDylanIvan()
{
    // Word we chose was candidate and we did our own way
    // of breaking the work up

    // I decided to break the word up into can, did, date
    // ,candid, didate (single number)

    //                       012345678
    String abc = new String("candidate");
    String word = " ";
    System.out.print("\n" + abc);

    word = abc.substring(0,3);
    System.out.print("\nword: " + word);
```

Class compiled - no syntax errors



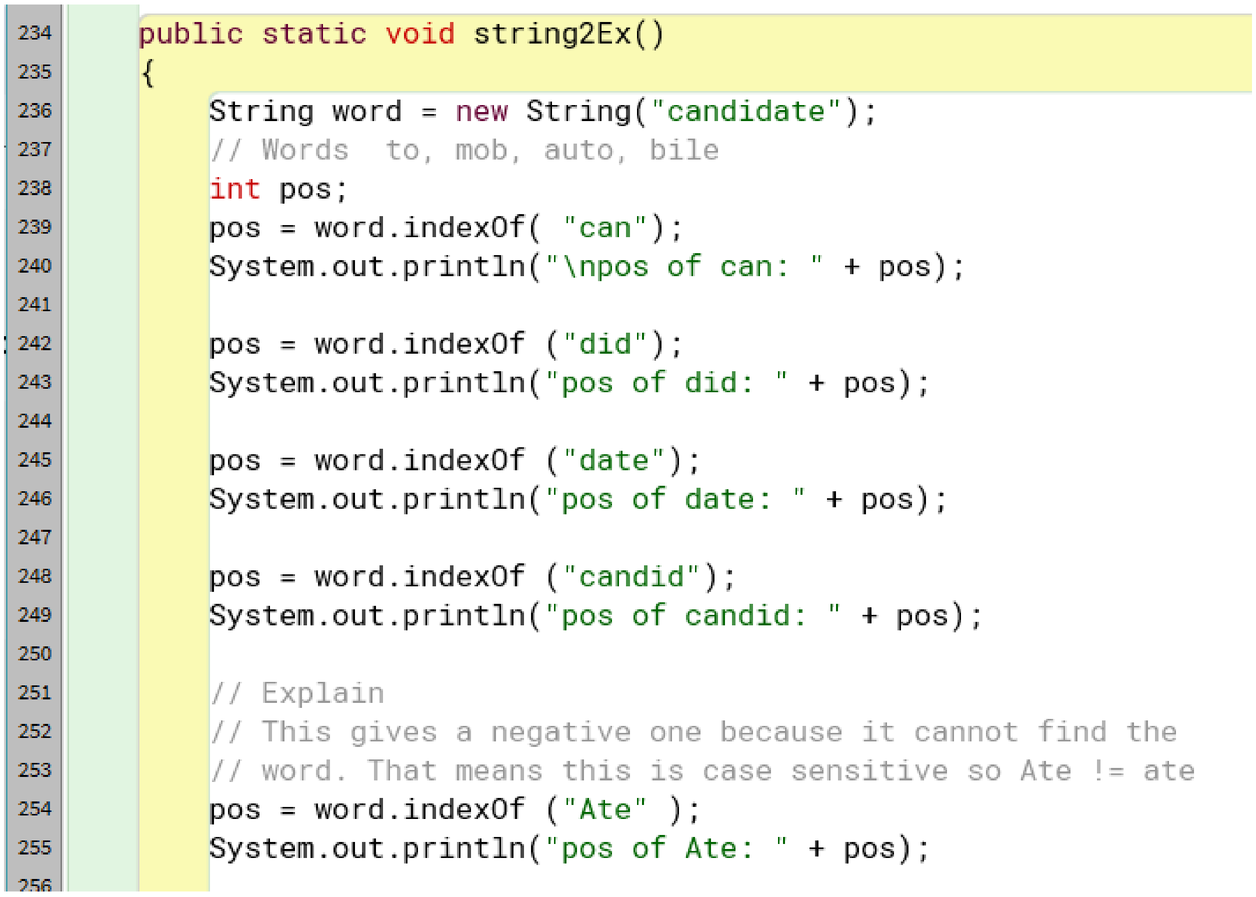
```java
        word = abc.substring(3,6);
        System.out.print("\nword: " + word);

        word = abc.substring(5,9);
        System.out.print("\nword: " + word);

        word = abc.substring(0,6);
        System.out.print("\nword: " + word);

        word = abc.substring(6);
        System.out.print("\nword: " + word);


    }
}
```

Results



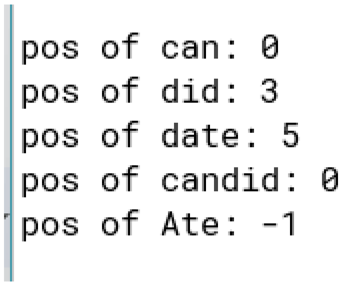
```
candidate
word: can
word: did
word: date
word: candid
word: didate
```

Took a break 4:00
We used substring to use the .equals method between strings



```java
    public static void string2Ex()
    {
        String word = new String("candidate");
        // Words  to, mob, auto, bile
        int pos;
        pos = word.indexOf( "can");
        System.out.println("\npos of can: " + pos);

        pos = word.indexOf ("did");
        System.out.println("pos of did: " + pos);

        pos = word.indexOf ("date");
        System.out.println("pos of date: " + pos);

        pos = word.indexOf ("candid");
        System.out.println("pos of candid: " + pos);

        // Explain
        // This gives a negative one because it cannot find the
        // word. That means this is case sensitive so Ate != ate
        pos = word.indexOf ("Ate" );
        System.out.println("pos of Ate: " + pos);
```

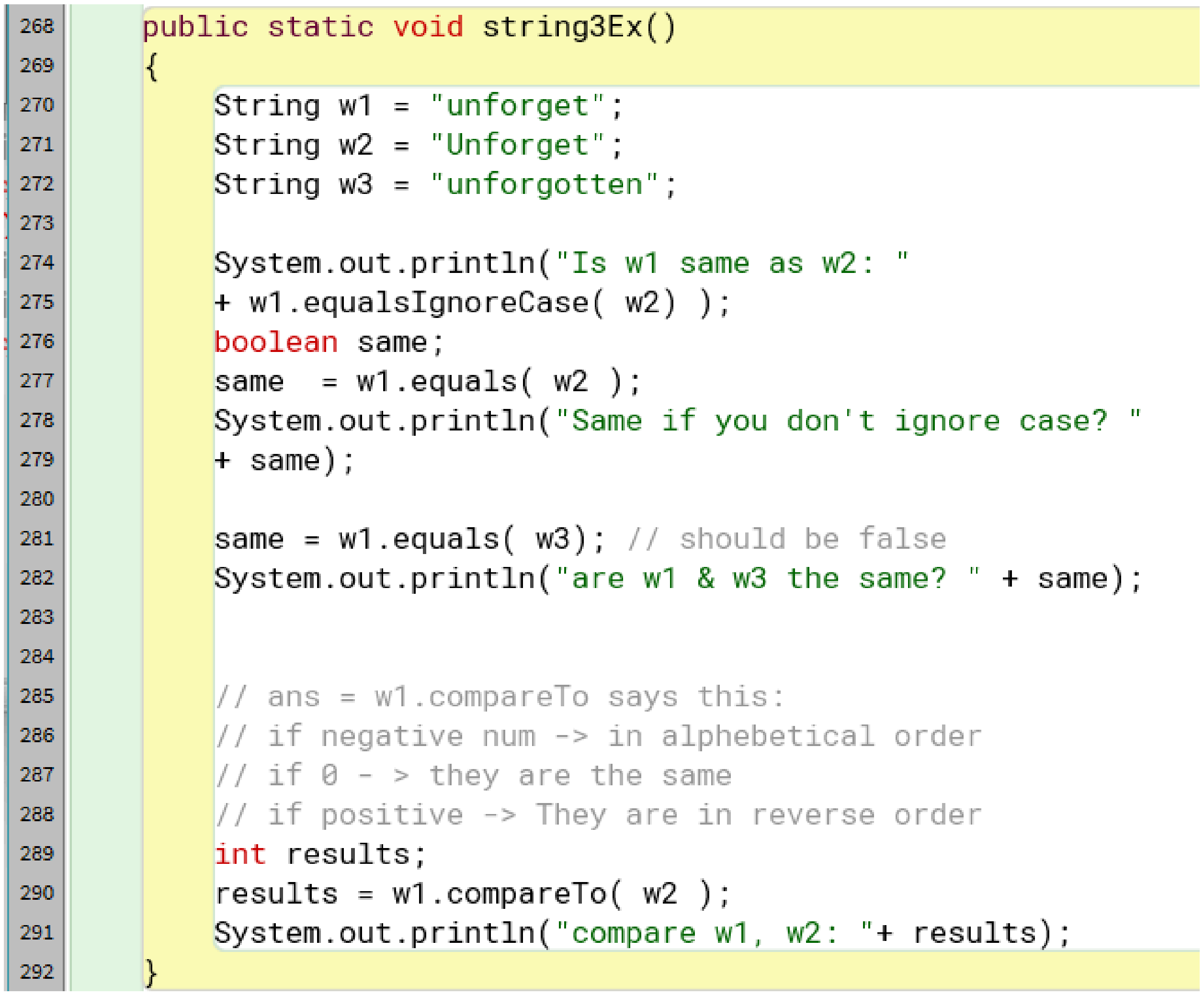
This gives us an index of where the word we are looking for starts



```
pos of can: 0
pos of did: 3
pos of date: 5
pos of candid: 0
pos of Ate: -1
```

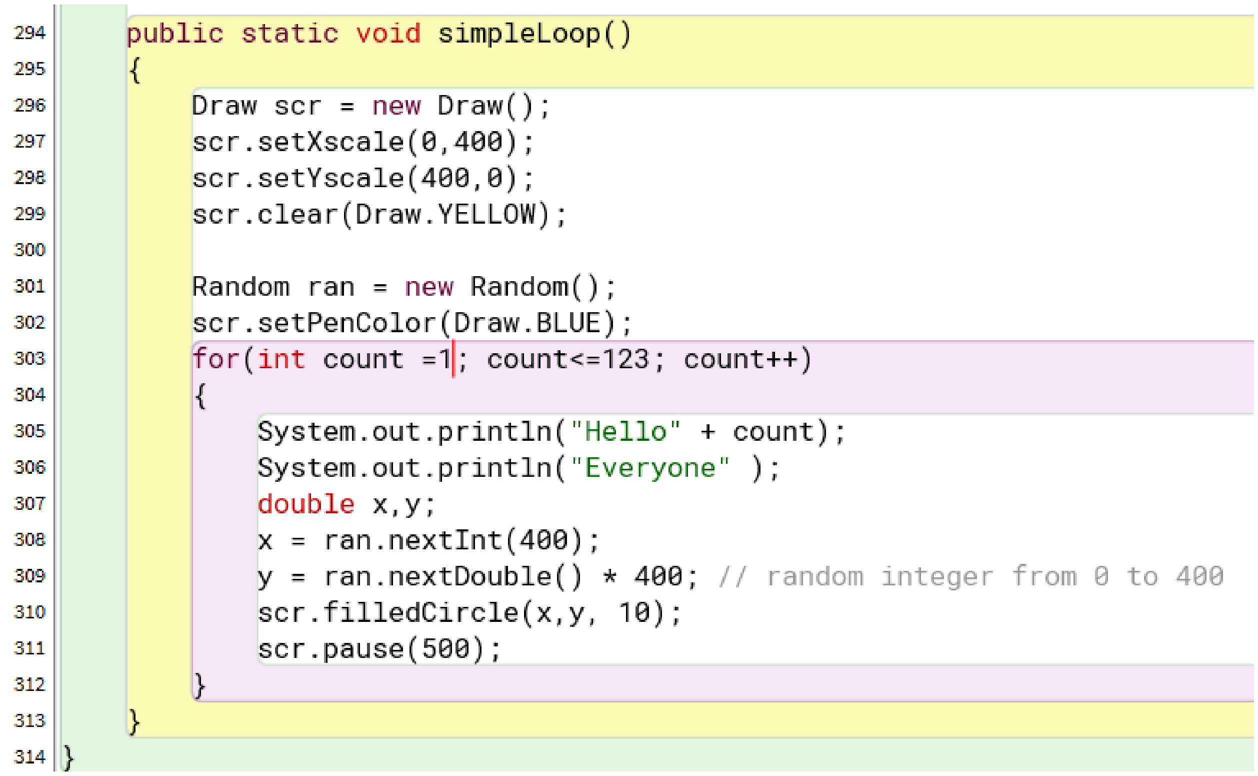
We can see here it gave back a negative 1 which means that it does not exist because the word we should be looking for is ate rather than Ate. The A is capitalized and a different value

Here we used the .equals method to compare similar words and the compareTo In the third string Example



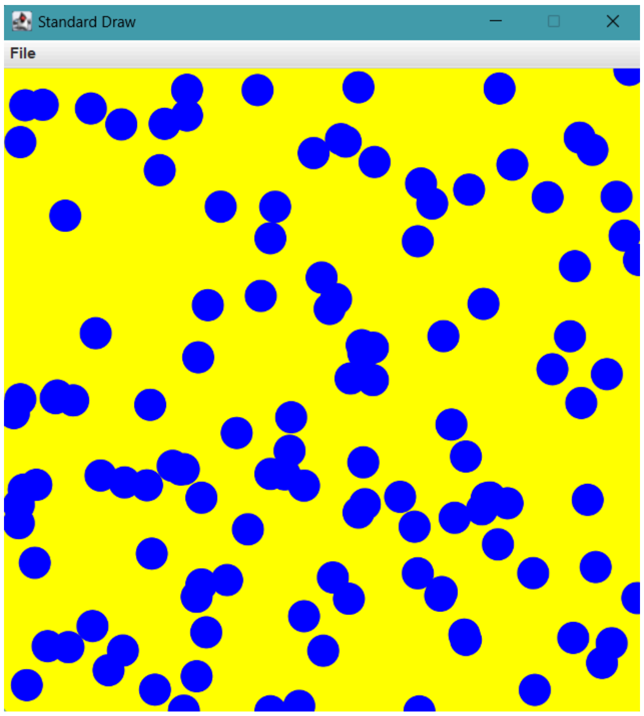
```java
public static void string3Ex()
{
	String w1 = "unforget";
	String w2 = "Unforget";
	String w3 = "unforgotten";

	System.out.println("Is w1 same as w2: "
	+ w1.equalsIgnoreCase( w2) );
	boolean same;
	same  = w1.equals( w2 );
	System.out.println("Same if you don't ignore case? "
	+ same);

	same = w1.equals( w3); // should be false
	System.out.println("are w1 & w3 the same? " + same);


	// ans = w1.compareTo says this:
	// if negative num -> in alphebetical order
	// if 0 - > they are the same
	// if positive -> They are in reverse order
	int results;
	results = w1.compareTo( w2 );
	System.out.println("compare w1, w2: "+ results);
}
```

Simple Loop



```java
    public static void simpleLoop()
    {
        Draw scr = new Draw();
        scr.setXscale(0,400);
        scr.setYscale(400,0);
        scr.clear(Draw.YELLOW);

        Random ran = new Random();
        scr.setPenColor(Draw.BLUE);
        for(int count =1; count<=123; count++)
        {
            System.out.println("Hello" + count);
            System.out.println("Everyone" );
            double x,y;
            x = ran.nextInt(400);
            y = ran.nextDouble() * 400; // random integer from 0 to 400
            scr.filledCircle(x,y, 10);
            scr.pause(500);
        }
    }
}
```

Output


Standard Draw
File