**Switch Statements:**

* Open Demos>Java>switch-statements in IntelliJ
* Open src>main>java>Main and show that we except an input of a number between 1 and 12 and print the corresponding month
* Show that we have a very long if statement to handle this functionality
* Rewrite the if statement into a switch statement but stop at June
* Hover over ‘switch’ and click ‘replace with switch expression’
* Complete the rest of the months as a switch expression

**Iteration Statements:**

* Open Demos>Java>loops in IntelliJ
* Open src>main>java>ForLoop.java
* Inside of fibonacciSequence() add a for loop that iterates from one to the value of count with the following body:
  + System.out.print(num1+" ");  
    int sumOfPrevTwo = num1 + num2;  
    num1 = num2;  
    num2 = sumOfPrevTwo;
  + After the loop print a new line
* In Main, set count to 7 and call ForLoop.fibonacciSequence(count)
* Show that it prints the first 7 numbers in the Fibonacci sequence
* In WhileLoop, add a variable for i set to 1 and create a while loop that iterates while i<=count
  + Give it the same body as the for loop
  + After the loop print a new line
* In Main, call WhileLoop.fibonacciSequence(count)
* Show that it prints the first 7 numbers in the Fibonacci sequence again
* In DoWhileLoop, add a variable for i set to 1 and create a do while loop that iterates while i<=count
  + Give it the same body as the for loop
  + After the loop print a new line
* In Main, call DoWhileLoop.fibonacciSequence(count)
* Show that it prints the first 7 numbers in the Fibonacci sequence again
* Change all the loops to iterate until i>=count and run the program again
* Now ForLoop and WhileLoop don’t print any numbers, but DoWhileLoop prints the first number in the sequence

**Interfaces:**

* Open Demos>Java>interfaces in IntelliJ
* Navigate to src>main>java and create an interface called Animal
* Create a method called makeAnimalSound(), a default methods called run() and eat(), all will return void
* Create a class called Dog that implements Animal and try to override all of the Animal interfaces methods
* You are not required to override run or eat because they have defaults, you are only required to override makeAnimalSound()
* Remove override of run and eat because dogs run and eat like most animals
* Create a Pig class and override all three methods
* In Main, try to instantiated a new animal – this will cause an error b/c interfaces cannot be instantiated
* Create a new Dog and a new Pig of type Animal
  + Dog dog – new Dog()
  + Animal pig = new Pig();
* Call all 3 methods on dog and pig