* git branch – list out all the branches
* Increment/decrement
  + Ex: int number = 125
    - System.out.println(++number);
    - Output: 126
  + Ex: int x = 14;
    - System.out.println(x++);
    - Output: 14
  + Ex: int x = 1;
    - Do {
      * System.out.println(“Do while loop…”)
    - } while (x < 5);
    - Output: None are correct
* LinkedList
  + Maintain insertion order
  + Uses a doubly linked list (data structure) to store the elements
  + Implements the List interface
* Set
  + Cannot contain duplicate values/elements
  + Classes:
    - Set<data-type> s1 = new HashSet<data-type>();
    - Set<data-type> s2 = new LinkedHashSet<data-tyep>();
    - Set<datat-type> s3 = new TreeSet<data-type>();
* HashSet
  + Contain unique items
  + Uses hash map for storage
  + Implements the set interface
* LinkedHash Set
  + Can contain unique elements
  + Maintains the insertion order
  + Permit null elements
  + Represents the LinkedList implementation of the set interface. It extends the HashSet class and implements the Set interface
* Map
  + Keys will contain unique keys
  + Contains values on the basis of key and value pairs
* HashMap
  + Implements the map interface
  + Contains only unique keys
  + May have one null key and multiple null values
  + Non synchronized
  + Maintains no order
* Iterator
  + An iterator is an object that allows for cycling through a collection to obtain or remove elements
  + The hasNext() method will return true if there is at least one more element
  + Provided class: hashNext() next() remove()
* OOP
  + OOP is a style that is intended to put programmers in the mindset of the real world
  + Java is not a fully OOP language
  + Each object has identity, attributes, behavior
* Class
  + When creating a class, the class name, and the java file which contains the class has to have the same name as the class
  + In order to create your own objects in java you must have a user defined class that you can instantiate to create an object
  + A class in java is the user defined blue print with fields and methods
* Attributes/fields
  + You can define as many fields as you want in a class
  + The attributes are the variables or fields within a class
  + Fields and methods of ex: legs, name, sound
* Access modifiers
  + Protected access modifier will allow access within the same package and it will also allow access to the child class if it resides outside the package
  + The private access modifier in java will bind the data to the class it resides in. Which means it is only accessible within that class
  + Public, private, protected, default
  + The public access modifier in java will allow you access to the data anywhere in the program
* Method
  + To call a method, type its name and then follow the name with a set of parentheses
  + Behavior in OOP
  + A method is a collection of statements that are grouped together to perform an operation
* Method overriding
  + To override a method the method must have the same method signature and parameters
  + You cannot override a static method
  + Method overriding is also known as run time polymorphism
* Method overloading
  + Another name for method overloading is compile time polymorphism
  + You can overload a static method
  + In the method must have the same name, but different parameters
* Constructor
  + You can overload a constructor
  + You cannot override a constructor
  + Constructors are special methods that are invoked after an object is created
  + A constructor must have the same name as the class it resides in
* Object
  + You can create an object by instantiating a class
  + Ex: Animal dog = new Animal();
* Static
  + When you create a static method, you can call that method by typing out the class name first and then the method name. Vehicle.horn()
  + The variable or method will belong to the class, rather than to a specific instance
* Final
  + Static and final can be use together
  + The final keyword is used to make a variable constant. This means that it will be assigned only once