Variables - 2

Non-Primitive / Reference data types

1. Class
    - Whenever we use 'new' keyword, we are creating new object in Heap
    - object created from Class is reference to the actual Heap memory
2. String
    - Inside Heap memory we have 'string constant pool'.
    - Whenever new string is created, it first checks 'string constant pool' for same literal, if present newly created string object will point to same literal.
    - When we use 'new' keyword, it is not considered string literal, it created another String object inside Heap and referenced to this object and not to string constant pool.
    - == , checks whether variables refer to same memory in Heap
    - .equals() - checks value inside that memory
    - Strings are immutable - once created they cannot be changed, if literal not present in 'string constant pool', new string literal will be created.
        - old string literal will still be there, just reference is changed
3. Interface
    - 'interface' keyword
    - when some class implements 'interface' it has to override that interface methods
    - An class which implements an interface, can store objects references of its own name or its parent implements interface
    - An object of an interface cannot be created - interface only defines blueprint not implementatio
    - default value = 0
4. Array
    - new int [6] | {0,9,0} - in Heap

- In java there is no concept of pointers
- In java everything is pass by value not pass by reference
- everything achieved in C with pointers, can be achieved with reference in Java

1. Wrapper Class
    - AutoBoxing
        - int a = 10;
        - Integer n = a; (primitive to wrapper class)
    - Unboxing
        - Integer x = 20;
        - int n = x; (wrapper class to primitive)
    - For each primitive type, we have their respective Reference type in Java (wrapper class)
    - if we want to pass primitive data type as Reference data type - we can wrap them inside wrappe classes.
    - Collections work on only reference data type
    -

2. Constant Variable - read only
    - 'static final' keyword - cannot be changed afterwards
    - name must be Capital