Unit 6 → Arrays

Arrays

- Array is an object that can store many values of the same type in a single variable
    - Can be a list of `Strings`, `ints` etc.
    - Stores a ***fixed number*** of elements of the same type in a single variable
    - SIZE ***CANNOT*** BE CHANGED
- Data type
- Declaration
    - Syntax → `Type [] variableName = new type[numberOfvalues];`
        - `Type [] variable Name = {values};`
    - Example →`int [] score = new int[5];`
        - `int [] score = {1 , 2 , 3 , 4 , 5};`

Type	**Default Value**
int	0
double	0.0
boolean	false
Object	null

- Getting values
    - Enter index of values
    - `[0]` gives the *first* value
    - Example → `int whatScore = score[0];`
        - Prints out → `1`
- Getting length of array
    - Syntax → `arrayName.length;`
    - Example → `int scoreLength = score.length;`
        - Prints → `5`

Transversing Arrays

- Transversing an array is to cycle through an array using a loop
- Syntax
    - `for (int i = 0; i < array.length; i++)`
    - `{`
        - `System.out.println(array[i]);`
    - `}`
- Iteration → amount of times the code runs
    - Usually equals to the `array.length` (during a `for` loop)
- `break;` stops the loop from continuing at the stop the line of code is

Enhanced For Loops

- An alternate method to transverse an array instead of using a `for` or `while` loop
    - Cannot exit while in-acton
- Also known as the For-Each Loop
- Efficent way to access objects
- Better used with nested loops
- Syntax
    - `for (int variable : array)`
    - `{`
        - *`//code goes here`*
    - `}`

Developing Algoritims Using Arrays

- Common array algorithms
    - Max and min value
        - Minimum
            - `int minIndex = 0;`
            - `for (int i = i; i < array.length; i++)`
            - `{`
                - `if (array[i] < array[minIndex])`
                - `{`
                    - `minIndex = i;`
                - `}`
            - `}`
    - Sum, average, or mode
        - For mode: use counter
        - Average
            - `for (int i = 0; i < array.length; i++)`
            - `{`
                - *`/calculation here`*
            - `}`
            - `return (double) sum / array.length;`

    - Determining properties of a particular property
        - Properties of a value
            - `int counter = 0;`

```
for (int i = 0; i < array.length -1; i++)
{
    if (array[i].equals("property goes here")
    {
        counter++;
    }
}
```

- Access consecutive pairs of elements
  - Check first number and if it is equal to the second, it's a consecutive pair
  - Consecutive

```
boolean consecutive = false;
for (int i = 0; i < array.length -1; i++)
{
    if (array[i] == array[i+1])
    {
        consecutive = true;
    }
}
```

- Reordering arrays
  - Shift or rotate elements left or right
  - Reverse order of elements
- Sorting arrays
  1. Start with a for loop on the first number
  2. Create a second for loop for the second number
  3. Take first number and compare to each number after
  4. If the first number is greater than the second, switch positions
  5. 2nd counter == 1st counter