Notes on Arrays in Java

Introduction to Arrays

1. Arrays are collections of similar data types stored in contiguous memory locations, accessible via indices.
2. Key Features:
    - Elements can be primitive or reference types.
    - Index starts from 0 to n-1.
    - Arrays are objects created dynamically at runtime.
    - Implement `Cloneable` and `Serializable` interfaces.
    - Can use `Object.clone()` for duplication and `Arrays.equals()` for comparison.

Types of Arrays

1. **Single-dimensional Array**:
    - A linear list of elements of the same type.
    - Accessed using a single index.
2. **Multi-dimensional Array**:
    - Arrays with more than one dimension, such as 2D or 3D arrays.
    - Accessed using multiple indices.

Array Declaration, Construction, and Initialization

1. **Declaration**:

    ```
    int[] a; // Recommended
    int []a;
    int a[];
    ```

    Size cannot be specified during declaration (e.g., `int[5] a;` is invalid).

2. **Construction**:

    - Arrays are objects created using the `new` keyword.
      Example: `int[] a = new int[3];`

3. **Initialization**:

    - Array elements are initialized to default values (e.g., `0` for `int`).
    - Elements can be assigned custom values:

    ```
    int[] a = new int[4];
    a[0] = 10;
    ```

4. **Single-line Initialization**:

    ```
    char[] ch = {'a', 'e', 'i'};
    ```

Key Rules

1. Size must be specified during construction: `int[] a = new int[3];`.
2. Array size can be `0` but not negative (throws `NegativeArraySizeException`).

3. Valid data types for size: `byte`, `short`, `char`, `int`. Others cause compile-time errors.
4. Maximum size is `Integer.MAX_VALUE` (2,147,483,647).

Special Array Concepts

1. **Anonymous Arrays**:
   - Arrays without a name, used for instant purposes.
     Example: `new int[] {10, 20, 30};`.
2. **length vs. length()**:
   - `length`: Array property for size.
     Example: `int[] a = new int[3]; System.out.println(a.length); // 3`
   - `length()`: String method for the number of characters.
     Example: `String s = "Java"; System.out.println(s.length()); // 4`.
3. **Jagged Arrays**:
   - Multi-dimensional arrays with varying row sizes.
     Example:

```
int[][] jagged = new int[3][];
jagged[0] = new int[]{1, 2};
jagged[1] = new int[]{3};
jagged[2] = new int[]{4, 5, 6};
```

Array Element Assignments

1. **Primitive Type Arrays**:
   - Element types can be promoted to declared types (e.g., `byte` → `int`).
   - Example:

```
int[] a = new int[3];
a[0] = 'A'; // Valid (char to int)
```

2. **Object Type Arrays**:
   - Can hold objects of declared types or their subclasses.
   - Example:

```
Object[] arr = new Object[3];
arr[0] = "Hello"; // Valid
arr[1] = 42;      // Valid
```

3. **Interface Type Arrays**:
   - Can hold objects of classes implementing the interface.
   - Example:

```
Runnable[] tasks = new Runnable[2];
tasks[0] = new Thread(); // Valid
```

Two-Dimensional Arrays

1. Declaration:

```
int[][] a;
int[] []a;
int a[][]; // All valid
```

2. Initialization:

```
int[][] a = {{1, 2}, {3, 4}};
```

3. Length property in 2D arrays:
   - `a.length` : Number of rows.
   - `a[0].length` : Number of columns in the first row.

MCQ Question Bank: Arrays in Java

General Concepts

1. **What is an array in Java?**
   a) A collection of objects of the same class
   b) A collection of values of the same data type stored in contiguous memory
   c) A dynamically resized list of elements
   d) A key-value pair data structure
   Answer: b

2. **Which of the following is true about arrays in Java?**
   a) Arrays are primitive data types.
   b) Arrays cannot store reference types.
   c) Arrays implement `Cloneable` and `Serializable`. d) Arrays are immutable in size and elements.
   Answer: c

3. **How are elements in a Java array indexed?**
   a) From 1 to n
   b) From 0 to n-1
   c) From -1 to n
   d) Based on user-defined indices
   Answer: b

Array Declaration, Construction, and Initialization

4. **Which of the following array declarations is valid?**
   a) `int[5] a;`
   b) `int a[];`
   c) `int[] a = new int[];`
   d) `int a[3] = new int[];`
   Answer: b

5. **What will happen if an array is created with a negative size?**
   a) It throws a `CompileTimeException`. b) It initializes to an empty array.
   c) It throws a `NegativeArraySizeException`. d) It results in undefined behavior.
   Answer: c

6. **What is the default value for elements in an `int` array?**
   a) Null
   b) 0
   c) Undefined
   d) -1
   Answer: b

7. **How can you declare, construct, and initialize an array in a single line?**
   a) `int[] a = new int(1, 2, 3);`
   b) `int[] a = {1, 2, 3};`
   c) `int a = {1, 2, 3};`
   d) `int[] a; a = {1, 2, 3};`
   Answer: b

Anonymous Arrays

8. **What is the primary purpose of anonymous arrays in Java?**
   a) To define multidimensional arrays
   b) To store null values temporarily
   c) To use arrays without declaring a name
   d) To initialize arrays at compile time
   Answer: c

9. **Which of the following creates a valid anonymous array?**
   a) `new int[3]{1, 2, 3};`
   b) `new int[]{1, 2, 3};`
   c) `new int[3] {1, 2};`
   d) `{1, 2, 3}`
   Answer: b

Array Properties and Features

10. **Which property is used to determine the size of an array in Java?**
    a) `size()`
    b) `length`
    c) `length()`
    d) `getSize()`
    Answer: b

11. **How does the `length` property differ from the `length()` method in Java?**
    a) `length` is for strings, `length()` is for arrays.
    b) `length` is a property for arrays, `length()` is a method for strings.
    c) Both are identical and interchangeable.
    d) `length()` can also be used with arrays.
    Answer: b

Array Assignments

12. **Which of the following types can be assigned to an `int` array?**
    a) `float`
    b) `long`
    c) `char`
    d) `double`
    Answer: c

13. **What happens when a parent class reference variable is assigned to a child class array?**
    a) Compile-time error
    b) Runtime exception
    c) Successful assignment
    d) All elements are copied automatically
    Answer: c

Multidimensional and Jagged Arrays

14. **What is a jagged array?**
    a) A 2D array with elements stored in non-contiguous memory locations.
    b) A multidimensional array with varying column sizes.
    c) An array that supports only primitive data types.
    d) An array that throws an exception for null values.
    Answer: b

15. **How can you declare a 2D array in Java?**
    a) `int[][] a = new int[2][3];`
    b) `int[][] a = new int[2][];`
    c) `int[][] a = {{1, 2}, {3, 4}};`
    d) All of the above
    Answer: d

16. **Which property gives the number of rows in a 2D array?**
    a) `a.length()`
    b) `a.getRows()`
    c) `a.length`
    d) `a.size`
    Answer: c

These MCQs cover the essential topics and nuances of arrays in Java, testing understanding of their declaration, initialization, features, and behavior in various contexts.