OOP GD

Harshal : Good afternoon to one and all present here today! We the members of group 3 will be discussing a very important and essential topic for object oriented programming that is Arrays and Collections.Arrays and Collection contribute as an important element when it comes to OOP data types as both have their own purpose and importance.So without any further delay lets see what actually these are and how to use them in our programs

Pratham: Yes Harshal, let's first discuss what actual arrays in OOP are and how we should use them to write an efficient JAVA program. So basically a Java **array** is an object which contains elements of a similar data type. Also, The elements of an array are stored in a contiguous memory location. It is a data structure where we store similar elements.

Uma: Yeah that's right, using Arrays we can store only a fixed set of elements. We can use Arrays to take an entire string of characters, such as a username, and chunk it out into single characters.

Atharv Sonawane: Yes Uma, also we can access the elements stored in Array by using the index number which by default starts from 0. It means that in order to access the first element of an array we have to access the 0th index position. We can also find out the size of an array using the Array length() function.

Manasi: Yeah so basically arrays are fun to deal with but why do we exactly need them. To answer this question let me elaborate on the need of arrays. We need arrays when we have a list of variables and we want to perform some comparison, calculation or function on all of them at once, When we want to recode all of them for instance reverse the numbering or such things, Also we use arrays when we want to look through a list of variables and find a specific value or create a flag when You have multiple records and you want to make a single record. Or we just want to sort or arrange elements in ascending or descending order.

Rucha: JAVA arrays offer a lot of uses and for the very same reason the The Java Language and the Java Virtual Machine directly support only one-dimensional arrays. Two-dimensional array are simulated with a one-dimensional array whose elements themselves are one-dimensional arrays. Higher dimensional arrays are constructed using a one-dimensional array. This approach allows great flexibility, and supports complicated data structures. Declaring arrays is a easy step in Java

Bhavin: Yes Rucha let me elaborate it. Declaration of array involves creating a reference variable and declaring the data type for that reference variable.
The syntax goes something like [modifiers] <data type> <refVariable>[]; Lets understand it with an example which is int array1[10] now this statement declares an integer array with size 10 and name as array1

Shashank: Yes so basically we can declare arrays like Bhavin said lets now discuss how to obtain the declared array, obtaining an array is a two-step process. First, you must declare a variable of the desired array type. Second, you must allocate the

memory that will hold the array, using new, and assign it to the array variable. We can access an array element by referring to the index number.Example:
String[] cars = {"Volvo", "BMW", "Ford", "Mazda"};

System.out.println(cars[0]);

Here, cars[0] indicates the first item of array which is Volvo

Let's further discuss the types of arrays that JAva supports, i.e Single Dimensional Arrays, then Multidimensional arrays.

Atharva Rathi: The single dimensional arrays also known as linear array, here elements are stored in a single row. For example if we creat simple array and pass 5 as its parameter then this would give an array of five elements. which are stored in a single line or adjacent memory locations.
In Java a multidimensional array is also known as array of arrays. We can say that A two dimensional array is an array of one dimensional arrays and a three dimensional array is an array of two dimensional arrays. It stores the data in rows and columns format.

Harshal: So far we discussed how to declare,access and what are the types of arrays. Now let's move ahead and discuss different types of operations that we can perform on an Array. Some of the basic operations that we perform on arrays are Traverse − Print all the elements in the array one by one.

- Insertion − Adds an element at the given index.
- Deletion − Deletes an element at the given index.
- Search − Searches an element in the array using the given index or the value.
- Update − Updates an element at the given index.

Shraddha: Yes Harshal we can perform alot of operations on Arrays and use them based on our needs. And for the same reason we can loop through the array elements with the for loop, and use the length property to specify how many times the loop should run. For the same reason we use the for-each loop. Not only this but we can also pass an array to a method in java inorder to reuse the same logic on any array. In addition to this JAVA also supports the concept of anonymous arrays so basically we don't need to declare the array while passing an array to the method. Isn't it the most flexible data type already!

Bhavin: Yes Shraddha indeed it is, but don't you guys think that working with arrays could sometimes cause an unwanted error or what we term it as an exception? Well yes it surely does but that also is well taken care of by the JVM as it throws an ArrayIndexOutOfBoundsException if length of the array is negative, equal to the array size or greater than the array size while traversing the array. Also we can call the getClass().getName() method on the array object for the purpose of tracing.

Chaitanya: So working with arrays is fun, Arrays have tremendous abilities and advantages over the regular data types as arrays support dynamic allocation of data. More advantages of arrays are

- Arrays enable multiple elements and values to be stored under a single name.

- The location of elements in arrays is extremely easy.
- Array occupies dynamic memory in which individual elements are stored in contiguous locations.
- Arrays can store a large amount of data that needs to be stored or analyzed.

Pratham: As like a coin has both sides so does JAVA arrays, There are also a few disadvantages of arrays

- The Java array size needs to be declared with a given array**.**
- The size of the array in Java also cannot be increased or decreased
- They can only store data of a single type.
- If arrays of a larger size than is required are declared, the memory may be wasted.

Uma: So i guess we have pretty much covered what actually Java Arrays are lets now move forward and discuss the JAVA collections. So yeah a collection is a structure that offers an architecture for storing and manipulating a collection of objects. Java Collections are capable of doing any data operations such as searching, sorting, insertion, manipulation, and deletion. A single object of objects in Java is referred to as a collection. Many interfaces such as Set, List, Queue, Deque and classes such as ArrayList, Vector, LinkedList, PriorityQueue, HashSet, LinkedHashSet, TreeSet are available in the Java Collection framework

Atharv Sonawane: Let's first see what an ArrayList is. The ArrayList class, which is part of the java.util package, is a resizable array.It is a List interface with a resizable array implementation. ArrayList allows all elements, including null, to be used in all optional list operations. This class includes methods to change the size of the array that is used internally to hold the list, in addition to implementing the List interface. We can say that this class is similar to Vector but without the synchronization.

Manasi: Yes that's right Atharv, I would like to add that there are two types of ArrayList: Single Dimensional ArrayList and two dimensional ArrayList. Single dimensional ArrayList is a collection framework component included in the java.util package. In Java, it offers us dynamic arrays. On the other hand Two dimensional arrays are in a matrix form it is distributed in rows and columns. This concept is similar to two-dimensional arrays. A two-dimensional ArrayList is similar to a single dimensional ArrayList, but it can be visualized as a grid with rows and columns.

Bhavin: ArrayList provides a wide range of advantages over normal arrays.

- We can add any type of data into ArrayList
- It allows multiple null values.
- Retrieval is faster in arraylist
- In ArrayList we can remove element also at specific index of ArrayList
- It allows to traversal in both direction

Although When a data entry is added to or withdrawn from an array-based list, the data must be shifted to keep the list up to date which is a major drawback. —Atharva R.

Harshal: Talking about collections, I can think of one more such structure popularly known as Linked List. The Collection foundation in the java.util package includes a

Linked List. This class implements the LinkedList data structure, which is a linear data structure in which the components are not stored in sequential order and each element is a distinct object having a data and address portion.
Pointers and addresses are used to connect the elements. Every element is referred to as a node. They are preferable over arrays because of their dynamic nature and ease of insertions and removals.

Pratham: I agree with you Harshal but i can list a few disadvantages of linked list.

- If the array is sorted, binary search may be used to find any element. However, even if the linked list is sorted, binary search is not possible, hence the complexity of searching in a linked list is O(n).
- Each element in the linked list requires additional memory space for the pointer.
- Cache locality is stronger in arrays than in linked lists, which can have a significant impact on speed.

Uma: While we talk about disadvantages of linked lists, We have another structure that overcomes these disadvantages which is HashSet. HashSet class in java implements the Set interface, which is backed by a table called HashMap. Also HashSet inherits the Abstract Set class and implements Set interface. Hashing mechanism is used to store the elements in the HashSet. It also permits the null elements and Insertion of elements is based on the hash code in HashSet as it doesn't support insertion order.

Manasi: Yes Uma but isn't it the fact that the hashset uses Hashmap to store objects in key-value pairs which may be accessed using a different form of index. A HashMap's keys and values are genuine objects. HashMap is like a legacy Hashtable class, but it is not synchronized and can store the null elements as well. Since Java 5, it is denoted as HashMap<K,V>, where K stands for key and V for value. Also It inherits the AbstractMap class and implements the Map interface.

Rucha: All right!, HashMap has several advantages due to the use of key-value pair such as

- A Hashmap allows you to input a key-value pair.
- It is a non-synchronized data structure.
- Also it is a fast iterator that never fails.

Bhavin: I agree with you guys but again i would like to add that all works well as long as there are different hash code values for each keys, but when When two separate keys give the same hashCode() value, the performance of the hashMap degrades, HashMap takes O(n) time. This is a major disadvantage how important factor iteration performance is

Shraddha: Yeah, so I guess we discussed what actually are collections. The collection framework as well as the different structures which are supported by the collection framework. So far we got to know about both the Arrays as well as the Collections and

how to use them in our program along with the benefits they provide. Let's have a quick review of the things we discussed . So basically we learnt that An Array is a collection of *indexed* and *fixed* numbers of *homogeneous* (same type) elements. The elements are stored in index The stored element position starts from '*zero' and second element position '1' and so on.*

Shashank: Also When we are defining an array we have to specify the size of the array. Example : String[] *students* = new String[10]; . Once we create an Array we can't increase the size, size is fixed in arrays. And Array elements are homogeneous, it means once we create an array we can store the same type of elements. For example in point 4 student arrays store only String type elements, because the array type is String.

Atharv Rathi: Next we discussed collections, The **Collection in Java** is a framework that provides an architecture to store and manipulate the group of objects.Java Collections can achieve all the operations that you perform on data such as searching, sorting, insertion, manipulation, and deletion. It has a hierarchy structure with many interfaces such as Iterable,Collection,List,Queue,Set and classes such as ArrayList,Linkedlist,Vector,Hashset and many more

Harshal: Also the Java collection supports many methods such as add,remove,addAll, removeAll,clear, size etc to work with different classes and interfaces. We choose collections over Arrays as Arrays are fixed in size, in development it's always not possible to define size of an array. We need to increase size based on requirement. Collections are not fixed in size, they are growable (size will be increased when it's reached max size).

Chaitanya: One more reason is Arrays can only hold one type of element. Collections can hold different types of elements. And also There is no default method support from arrays for sorting, searching, retrieving etc… but collections have default utility method support, it's handy for a developer as it reduces the coding time.

Manasi: You guys are right, there are various factors on which the arrays and collection differ from each other. The first such factor is the size as mentioned earlier Arrays are fixed in size whereas the The collection is dynamic in size i.e based on requirement size could be altered even after its declaration. Another such factor is the data type support, Arrays can hold only the same type of data. Collection, on the other hand, can hold both homogeneous and heterogeneous elements.

Rucha: I can think of one more point which is the memory consumption Arrays due to fast execution consumes more memory and has better performance. Whereas on the other hand Collections consume less memory but also have low performance as compared to Arrays.

Shashank: Also Rucha, the performance here is an important factor on which they differ from each other. Arrays due to its storage and internal implementation are better in performance.Collection on the other hand with respect to performance is not recommended to use if we wish to achieve fast execution.

Atharv Sonawane: So to sum up the discussion, All the above limitations of array and advantages of Collections over arrays we can think of using Collections everytime we write some program. But that's not the case, We can't store primitive types in Collections, also performance wise arrays are better. It is recommended that when we know the size in advance then arrays are better to use, but if size is not fixed and data processing like sorting, searching is required then surely implementing collections is a smart choice.

Atharv Rathi: So to conclude our discussion, Each of the both have their own advantages and disadvantages. It is upto us the developers to choose from these options and develop not only working but a smart code/program that would speed up the execution, would take less memory space and would outperform its alternative solutions.

Shraddha: So yes I guess we have covered almost all the points which were important to understand the concept of Arrays, Collection and the major differences between the both to make a smart choice when we are to develop a better program. I thank everyone present here for allowing us to discuss such an important topic and with this I conclude our group discussion Thank you!