Java.

Assignment 1.

Q.1 What is the default value of Array for different data type?

→ Arrays are collections of elements, where each element can be of a specific data type. The default value of elements in an array often depends on the programming ~~lat~~ language and data type of Array.

java:

- 'int' array: 0 for each element
- 'float array: 0.0 for each element
- 'char' array: '\U0000' (null character) for each element
- 'boolean' array: 'false' for each element.
- object arrays: 'null' for each element

Q.2 Can you Pass the negative number in Array size?

→ No, you cannot create an array with a negative size. Attempting to do so will result in a runtime exception. Java programming language enforces non negative array size to ensure memory safety & prevent undefined behaviour.
for ex.

```
public class NegativeArray {
    public static void main(String[] args)
    {
        int[] arr = new arr[-2];
    }
}
```

Q.3 Where does Array Stored in JVM. Memory.

→ In Java Virtual Machine arrays are stored in heap Memory.
The heap is region of memory dedicated to storing objects (including Arrays) that dynamically allocated during program execution.

When you create an array in Java using 'new' keyword JVM allocates Memory. JVM allocates Memory for array object on Heap.

Q.4 What is an Anonymous Array in Java & Give an example?

→ An Anonymous array in Java is an array that is created without explicity assigning it a variable name. It is used in situation where you need a temporary array for specific purpose.
Anonymous Arrays are created using array initialization Syntax directly within method argument or expression.

Example of Anonymous Array

```
Public class AnonymousArray {
    public static void main (String [] args)
    {
        int sum = CalculateSum (new int [] {10,20,30,40,50});
        System.out.println ("Sum of elements :" + sum);
```

```
for (String name : new String[] { "Rahul", "Hari",
        "Nitin", "charlie" } )
    { System.out.println ("Hello" +name);
}

Public static int CalculateSum (int [] array) {
        int Sum=0;
        for (int num : array) {
            Sum+=num;
        }
    return Sum;
}
}
```

Q.5) What is disadvantages of Arrays ?

→ i) Fixed Size :- Arrays have fixed size which mean you need to know the number of elements you want to store in advance.

ii) Memory Allocation : Arrays are allocated a contigeous block of memory. which can lead to Memory wastage if array size is larger than necessary.

iii) Index Bounds checking : Java arrays do not perform automatic bound checking.

Q.6) What are the different ways to traverse an Array in Java?

→ In Java their are several ways to traverse (iterate through) an array. Here are some common Methods:

1. For Loops :- The Traditional for loop can be used to iterate over each element of an array. it provide ~~precice~~ presice control over the loop & works well for arrays of known size

ex

```
for (int i=0; i<array.length; i++){

}
```

2. Enhanced for loop (foreach loop) :- this loop Simplifies the Syntax for iterating over arrays, Collections, or other iterable objects. It is useful when you don't need the index value.

```
for (int element : array) {

}
```

3. While loop :- We can use while loop with an index to traverse an array.

ex :-
```
int i=0;
while (i<arr.length){
    System.out.println(arr[i]);
} i++;
}
```

Q.7 What is the difference between length & length() method Give an Example?

→

In Java. 'length' and 'length()' are used to obtain size or length of different types of data structures but they are used in different contexts and for different data types.

1. length (for arrays)

length property used for find length (number of elements) of array. it is public final instance variable of arrays.

```
int[] arr = {1,2,3,4,5};
int arraylength = arr.length;
System.out.println("Array length:" + arraylength);
```

2 length() (for Strings & other classes.

The length() method is used to find length (number of characters) of a String. It is a method ~~property~~ provided by String class

Ex:-

```
String str = "Hello"
int strlength = str.length();
System.out.println("String length" + strlength);
```