**DAILY ASSESSMENT FORMAT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **10-06-2020** | **Name:** | **Kiran N** |
| **Course:** | **JAVA** | **USN:** | **4al16ec031** |
| **Topic:** | **Programming core java**  **1.Arrays of Strings**  **2.Multi-Dimensional Arrays**  **3.Classes and Objects**  **4.Methods**  **5.Getters and Return Values**  **6.Method Parameters**  **7.Setters and "this"**  **8.Constructors**  **9.Static (and Final)**  **10.String Builder and String**  **Formatting** | **Semester & Section:** | **8th and A** |
| **Github Repository:** | **Kiran-course** |  |  |

|  |
| --- |
| **FORENOON SESSION DETAILS** |
| **Programming**    Java String array is used to hold fixed number of Strings.  String array is very common in simple java programs , specially among beginners to java and to test some specific scenarios. Even java main method argument is string array – public static void main(String[] args). So today we will look into different aspects of java string array with example programs.  Java String array is basically an array of objects.  There are two ways to declare string array – declaration without size and declare with size.  There are two ways to initialize string array – at the time of declaration, populating values after  declaration.  We can do different kind of processing on string array such as iteration, sorting, searching etc.  Java String Array Declaration  Below code snippet shows different ways for string array declaration in java.  String[] strArray; //  Declare without size String [] strArray1 =new String [3]; //declare with size  Note that we can also write string array as String strArray[] but above shows way is the standard and recommended way. Also in the above code, strArray is null whereas strArray1 value is [null, null, null].  Java String Array Initialization Let’s look at different ways to initialize string array in java. //inline initialization  String[] strArray1 = new String [] {"A","B","C"};  String[] strArray2 = {"A","B","C"};  //initialization after declaration  String[] strArray3 = new String[3];  strArray3[0] = "A";  strArray3[1] = "B";  strArray3[2] = "C";  Multidimensional Arrays in Java  Array-Basics in Java  Multidimensional Arrays  can be defined in simple words as array of arrays. Data in multidimensional arrays are stored in tabular form (in row major order).  Syntax:  data\_type  [1st dimension][2nd dimension][]..[Nth dimension] array\_name = new data\_type [size1][size2] ....[sizeN];  where:  •data\_type: Type of data to be stored in the array. For example: int, char, etc.  •dimension: The dimension of the array created.  For example: 1D, 2D, etc.  •array\_name: Name of the array  •size1, size2, ..., sizeN: Sizes of the dimensions respectively.  Examples:  Two dimensional array: int[][] twoD\_arr = new int[10][20];  Three dimensional array: int[][][] threeD\_arr = new int[10][20][30];  Classes and Objects in Java  Classes and Objects are basic concepts of Object Oriented Programming which revolve around the real life entities.  Class  A class is a user defined blueprint or prototype from which objects are created.  It represents the set of properties or methods that are common to all objects of one type. In general, class declarations can include these components, in order:  1.Modifiers: A class can be public or has default access (Refer this for details).  2.Class name:The name should begin with a initial letter (capitalized by convention).  3.Superclass(if any):The name of the class’s parent (superclass), if any, preceded by the keyword  extends. A class can only extend (subclass) one parent.  4.Interfaces(if any): A comma-separated list of interfaces implemented by the class, if any, preceded by the keyword implements. A class can implement more than one interface.  5.Body: The class body surrounded by braces, { }.  Object  It is a basic unit of Object Oriented Programming and represents the real life entities.  A typical Java program creates many objects, which as you know, interact by invoking methods. An object consists of :  1.State:It is represented by attributes of an object. It also reflects the properties of an object.  2.Behavior:It is represented by methods of an object. It also reflects the response of an object  with other objects.  3.Identity:It gives a unique name to an object and enables one object to interact with other  objects.  Example of an object : dog  String Formatting  The most common way of formatting a string in java is using String.format(). If there were a “java  sprintf” then this would be it.  String output=String .  format  ("%s = %d", "joe", 35);  For formatted console output, you can use printf() or the format() method of System.out and System.err PrintStreams. System . out .printf  (  "My name is: %s%n" , "joe");  Create a Formatter and link it to a StringBuilder.  Output formatted using the format() method will be appended to the StringBuilder .StringBuilder  Sbuf =new StringBuilder();  Formatter fmt= new Formatter(sbuf);  fmt.  format("PI = %f%n", Math.PI);  System.out.print  (sbuf.toString());  // you can continue to append data to sbuf here |