**DAILY ASSESSMENT FORMAT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **10 June 2020** | **Name:** | **Shreya poojary** |
| **Course:** | **java** | **USN:** | **4al16ec074** |
| **Topic:** | **Arrays of Strings**  **Classes and Objects** | **Semester & Section:** | **8-B** |
| **Github Repository:** | **Shreya-test** |  |  |

|  |
| --- |
|  |
| **Image of session** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  | |
|  | | | | | |
| **REPORT**  A Class is like an object constructor, or a "blueprint" for creating objects. Create a Class To create a class, use the keyword class: MyClass.javaCreate a class named "MyClass" with a variable x:publicclassMyClass{int x =5;}Create an Object In Java, an object is created from a class. We have already created the class named MyClass, so now we can use this to create objects.  To create an object of MyClass, specify the class name, followed by the object name, and use the keyword new:  Example  Create an object called "myObj" and print the value of x:  publicclassMyClass{  int x =5;  publicstaticvoidmain(String[]args){  MyClassmyObj=newMyClass();  System.out.println(myObj.x);  }  }  **Multiple Objects**  You can create multiple objects of one class:  Example  Create two objects of MyClass:  publicclassMyClass{  int x =5;  publicstaticvoidmain(String[]args){  MyClassmyObj1=newMyClass();// Object 1  MyClassmyObj2=newMyClass();// Object 2  System.out.println(myObj1.x);  System.out.println(myObj2.x);  }  }  A Java constructor is special method that is called when an object is instantiated. In other words, when you use the new keyword. The purpose of a Java constructor is to initializes the newly created object before it is used. This Java constructors tutorial will explore Java constructors in more detail.  Here is a simple example that creates an object, which results in the class constructor being called:  MyClassmyClassObj = new MyClass();  This example results in a new MyClass object being created, and the no-arg constructor of MyClass to be called. You will learn what the no-arg constructor is later.  A Java class constructor initializes instances (objects) of that class. Typically, the constructor initializes the fields of the object that need initialization. Java constructors can also take parameters, so fields can be initialized in the object at creation time.  **Constructor**  Here is a simple Java constructor declaration example. The example shows a very simple Java class with a single constructor.  public class MyClass {  public MyClass() {  }  }  The constructor is this part:  public MyClass() {  }  The first part of a Java constructor declaration is an access modifier. The access modifier have the same meanings as for methods and fields. They determine what classes can access (call) the constructor. | | | | | |
|  | | | |