**DAILY ASSESSMENT FORMAT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **11 June 2020** | **Name:** | **Shreya poojary** |
| **Course:** | **java** | **USN:** | **4al16ec074** |
| **Topic:** | **The Package to String Method**  **Inheritance** | **Semester & Section:** | **8-B** |
| **Github Repository:** | **Shreya-test** |  |  |

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

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | |  | |  | | |
|  | | | | | | | | | |
|  | |  | |  | |  | | | | |
|  | |  | |  | |  | | | | |
|  | |  | |  | |  | | | | |
|  | | | | | | | | |
| **REPORT** toString() Method The method is used to get a String object representing the value of the Number Object.  If the method takes a primitive data type as an argument, then the String object representing the primitive data type value is returned.  If the method takes two arguments, then a String representation of the first argument in the radix specified by the second argument will be returned. Syntax Following are all the variants of this method −  String toString()  static String toString(int i) Parameters Here is the detail of parameters −   * **i** − An int for which string representation would be returned.  Return Value  * **toString()** − This returns a String object representing the value of **this** Integer. * **toString(int i)** − This returns a String object representing the specified integer.  Example publicclassTest{  publicstaticvoidmain(Stringargs[]){  Integer x =5;  System.out.println(x.toString());  System.out.println(Integer.toString(12));  }  }  This will produce the following result − Output 5  12 Inheritance in Java **Inheritance in Java** is a mechanism in which one object acquires all the properties and behaviors of a parent object. It is an important part of [OOPs](https://www.javatpoint.com/java-oops-concepts) (Object Oriented programming system).  The idea behind inheritance in Java is that you can create new [classes](https://www.javatpoint.com/object-and-class-in-java) that are built upon existing classes. When you inherit from an existing class, you can reuse methods and fields of the parent class. Moreover, you can add new methods and fields in your current class also.  Inheritance represents the **IS-A relationship** which is also known as a parent-child relationship. The syntax of Java Inheritance  1. **class** Subclass-name **extends** Superclass-name 2. { 3. //methods and fields 4. }   The **extends keyword** indicates that you are making a new class that derives from an existing class. The meaning of "extends" is to increase the functionality.  In the terminology of Java, a class which is inherited is called a parent or superclass, and the new class is called child or subclass. | | | | | | | | |
|  | | | | | | | | | |
|  | | | | | | |