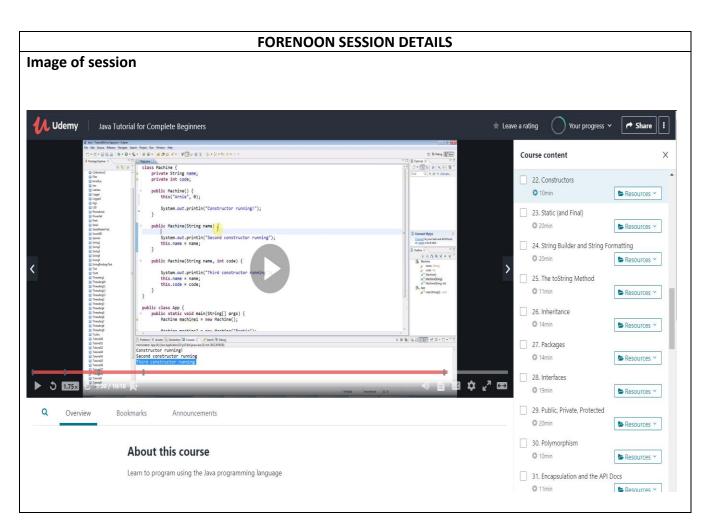
DAILY ASSESSMENT REPORT

Date:	11 June 2020	Name:	Gagan M K
Course:	Java Tutorial for Complete Beginners	USN:	4AL17EC032
Topic:	 The toString Method Inheritance Packages Interfaces Public, Private, Protected Polymorphism Encapsulation and the API Docs Casting Numerical Values Upcasting and Downcasting Using Generics 	Semester & Section:	6 th sem & 'A' sec
GitHub Repository:	Alvas-education- foundation/Gagan-Git		



Report – Report can be typed or hand written for up to two pages.

Java Programming:

- If you want to represent any object as a string, toString() method comes into existence
- The toString() method returns the string representation of the object.
- Inheritance is a mechanism in which one object acquires all the properties and behaviors of a parent object. It is an important part of OOPs.
- Encapsulation in Java is a mechanism of wrapping the data (variables) and code acting on the data (methods) together as a single unit.
- A generic method's body is declared like that of any other method. Note that type
 parameters can represent only reference types, not primitive types (like int, double and
 char).
- Type casting is when you assign a value of one primitive data type to another type.
- Upcasting (Generalization or Widening) is casting to a parent type in simple words casting
 individual type to one common type is called upcasting while downcasting (specialization
 or narrowing) is casting to a child type or casting common type to individual type.
- Java Generic methods and generic classes enable programmers to specify, with a single method declaration, a set of related methods, or with a single class declaration, a set of related types, respectively.

```
public class App {
    public static void main(String[] args) {
        Frog frog1 = new Frog(7, "Freddy");
        Frog frog2 = new Frog(5, "Roger");

        System.out.println(frog1);
        System.out.println(frog2);
}
```

• Inheritance Example:

```
public class Car extends Machine {

@ @Override
    public void start() {
        System.out.println("Car started");
    }

@ public void wipeWindShield() {
        System.out.println("Wiping windshield");
    }

@ public void showInfo() {
        System.out.println("Car name: " + name);
    }
}
```

Polymorphism example:

```
public class App {

   public static void main(String[] args) {
      Plant plant1 = new Plant();
      Tree tree = new Tree();

   Plant plant2 = tree;
   plant2.grow();
      tree.shedLeaves();

      //plant2.shedLeaves();

      doGrow(tree);
   }

   public static void doGrow(Plant plant) {
      plant.grow();
   }
}
```

Note:

• Today's "PCB Design" Session was completed yesterday and has been updated in Yesterday's report i.e. on 10 June 2020