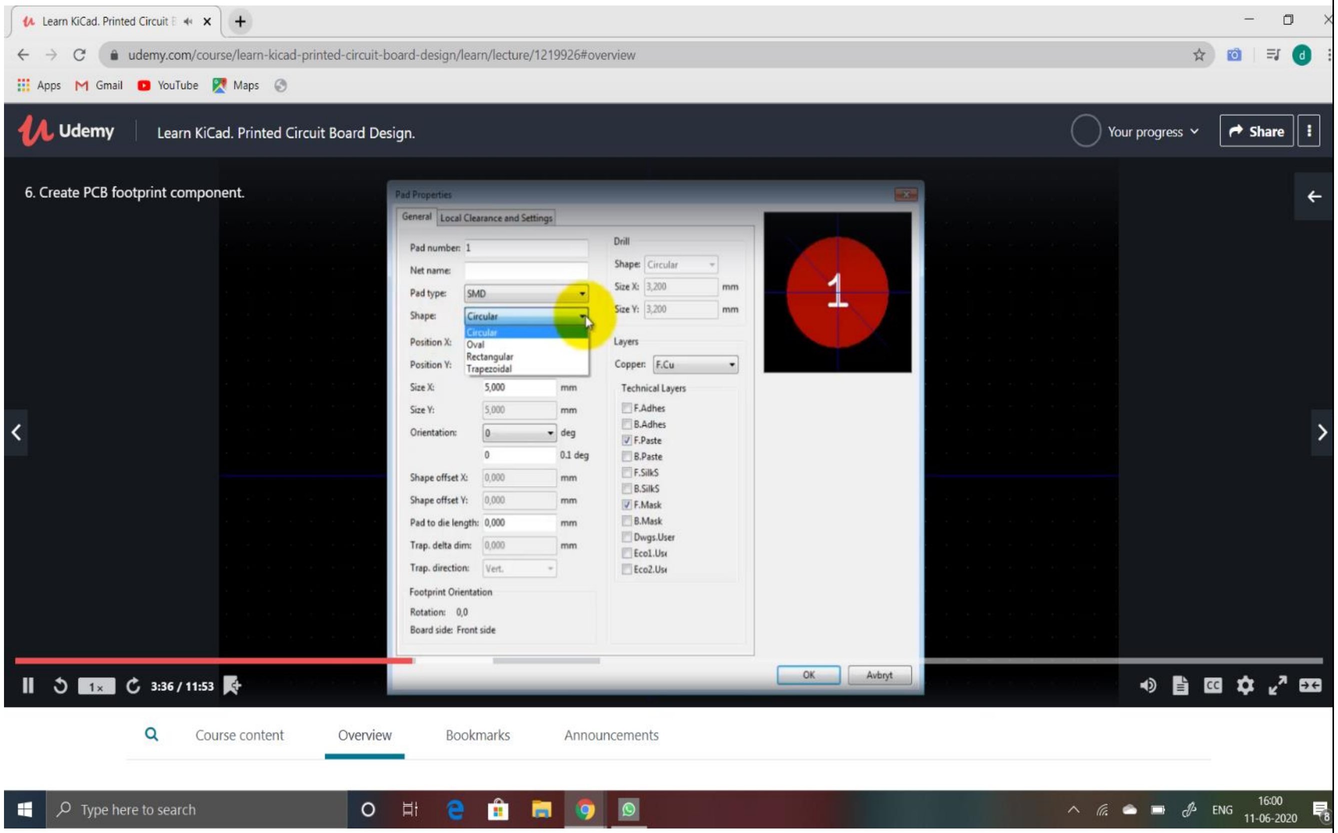
<u>DAILY ASSESSMENT FORMAT</u>

Date:	**11/06/2020**	**Name:**	**Yashaswini.R**
Course:	**PCB design using Kicad**	**USN:**	**4AL17EC098**
Topic:	**1. Creating Your Library 2. Create PCB footprint component.**	**Semester & Section:**	**6th sem 'B' sec**
Github Repository:	**Yashaswini**		

FORENOON SESSION DETAILS

Image of session



Learn KiCad. Printed Circuit Board Design.

6. Create PCB footprint component.

Pad Properties

General | Local Clearance and Settings

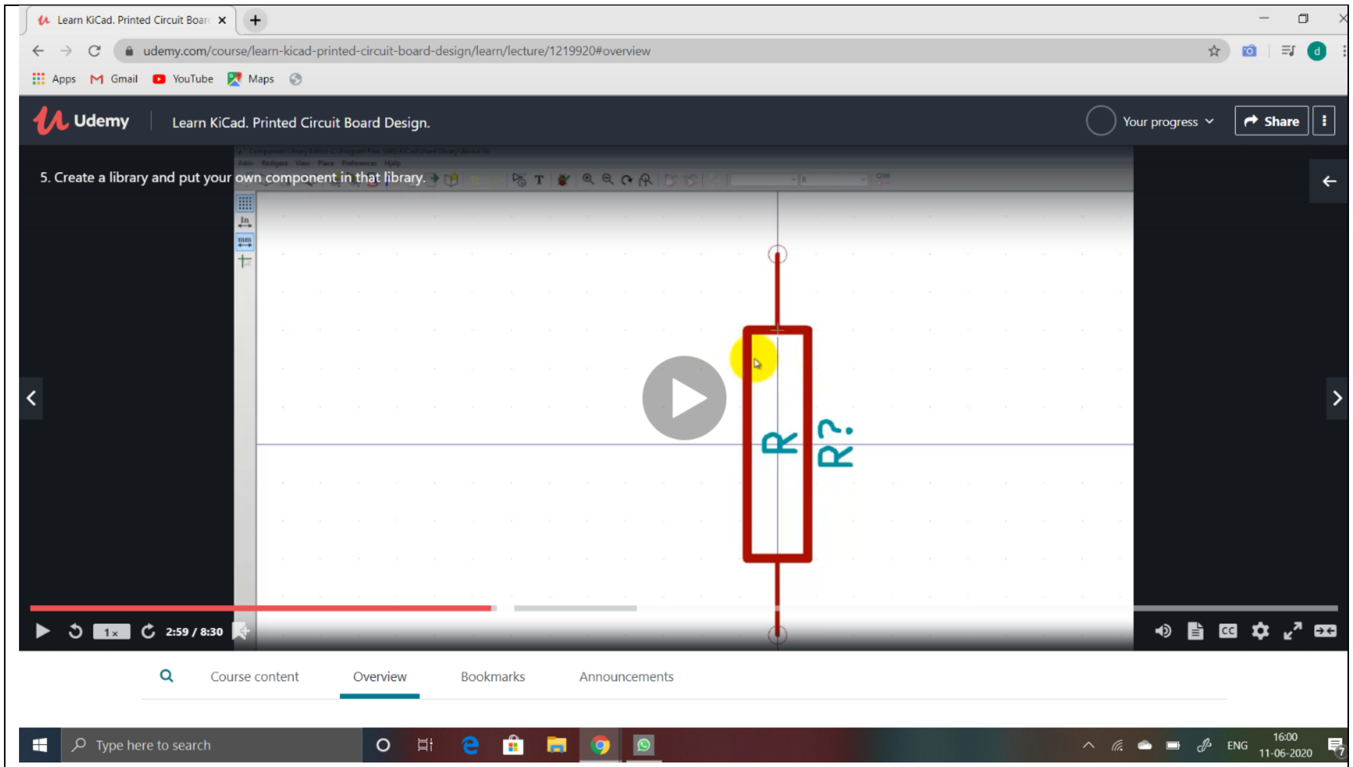
Pad number: 1
Net name:
Pad type: SMD
Shape: Circular (Circular, Oval, Rectangular, Trapezoidal)
Position X:
Position Y:
Size X: 5,000 mm
Size Y: 5,000 mm
Orientation: 0 deg
0 0.1 deg
Shape offset X: 0,000 mm
Shape offset Y: 0,000 mm
Pad to die length: 0,000 mm
Trap. delta dim: 0,000 mm
Trap. direction: Vert.

Footprint Orientation
Rotation: 0,0
Board side: Front side

Drill
Shape: Circular
Size X: 3,200 mm
Size Y: 3,200 mm

Layers
Copper: F.Cu

Technical Layers
- [ ] F.Adhes
- [ ] B.Adhes
- [x] F.Paste
- [ ] B.Paste
- [ ] F.SilkS
- [ ] B.SilkS
- [x] F.Mask
- [ ] B.Mask
- [ ] Dwgs.User
- [ ] Eco1.User
- [ ] Eco2.User


Learn KiCad. Printed Circuit Boar
udemy.com/course/learn-kicad-printed-circuit-board-design/learn/lecture/1219920#overview
Udemy
Learn KiCad. Printed Circuit Board Design.
Your progress
Share
5. Create a library and put your own component in that library.
R
R?
2:59 / 8:30
Course content
Overview
Bookmarks
Announcements
Type here to search

1.Creating Your Library

Open KICAD, and we'll be greeted with the **Control Panel**.

1. Next, select **File » New Library** to open **Library Window**.
2. Before creating any new parts, you need to save library by selecting **File » Save** (or **Cmd + S** on Mac and **Ctrl + S** on Windows)
3. Now that your library is saved, you just need to activate it. Go back to **Control Panel**, right-click your new library, and select **Use**.

2.Create PCB footprint component.

- **The blank canvas of the new footprint editor.**

You can also launch the footprint editor from the main Kicad window. To do this, go to the main Kicad window and click on the fourth button from the right. Notice that it contains the same icon as the one in Eeschema, with the IC and a pencil over it.

- **Creating the new footprint. We need to add boundaries and pins.**

Move the two text labels so that there is enough space between them for the boundary. Use the M key for this, just like you moved components in Eeschema.

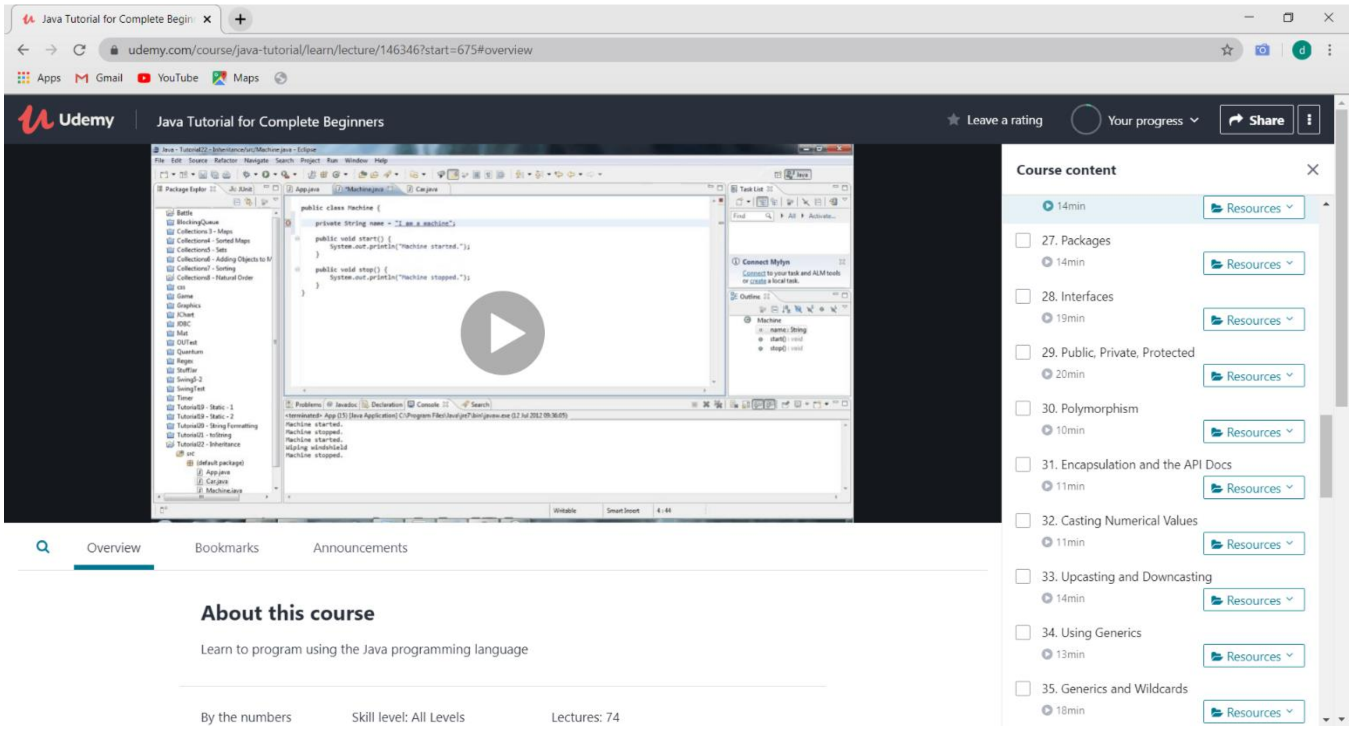
Let’s continue with the holes for the pins. We need eight holes. We need to select holes that are wide enough so that the pins of the actual component will fit through them and that the exact distance from the adjoining pins as we measured earlier. So, let’s start with a pitch. We know that the pitch (distance) between the pins is 2.54 millimeters, so to make it easy to space the pins at this exact distance, we’ll set the grid to be this size.

DAILY ASSESSMENT FORMAT

Date:	**11/06/2020**	**Name:**	**Yashaswini.R**
Course:	**Java**	**USN:**	**4AL17EC098**
Topic:	**Programming core java**	**Semester & Section:**	**6th sem 'B' sec**
Github Repository:	**Yashaswini**		

AFTERNOON SESSION DETAILS

Image of session



```java
public class Machine {

    private String name = "I am a machine";

    public void start() {
        System.out.println("Machine started.");
    }

    public void stop() {
        System.out.println("Machine stopped.");
    }
}
```

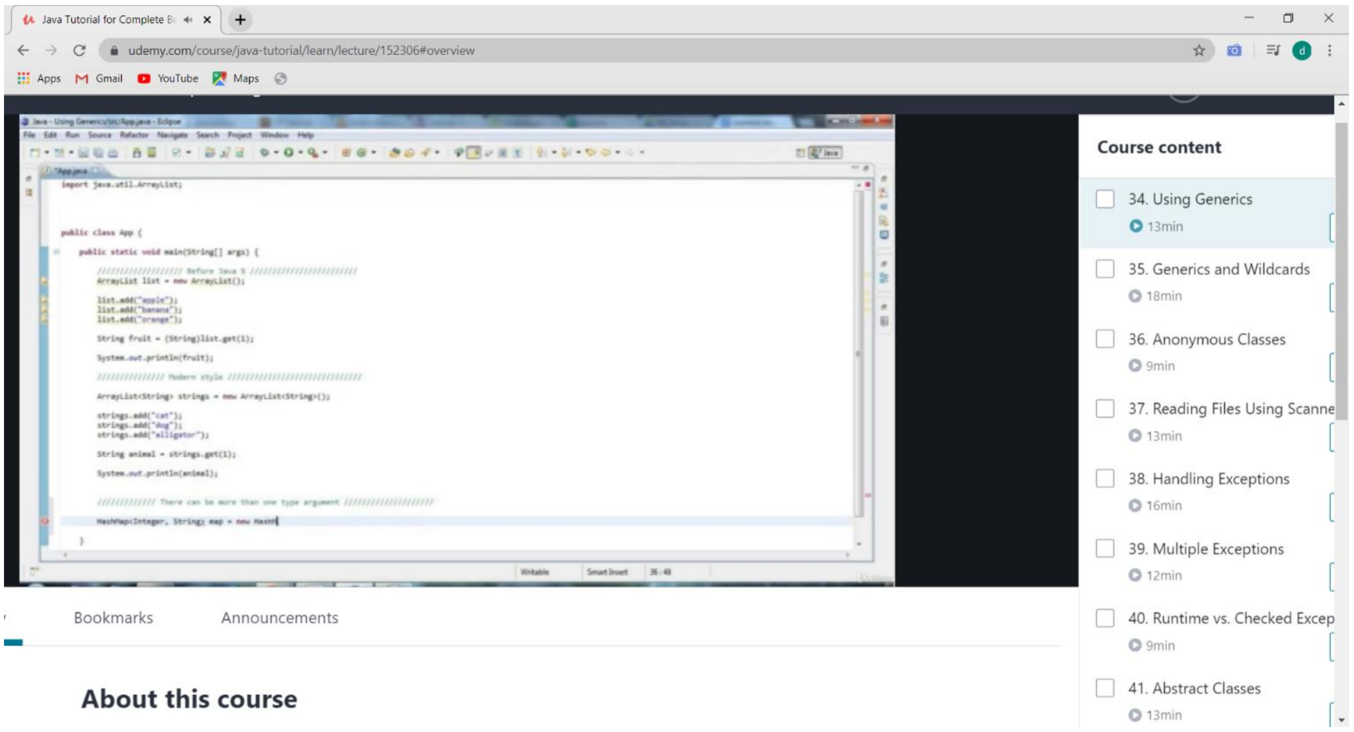
```
<terminated> App (15) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (12 Jul 2012 09:36:05)
Machine started.
Machine stopped.
Machine started.
Wiping windshield
Machine stopped.
```

Java Tutorial for Complete Beginners
udemy.com/course/java-tutorial/learn/lecture/146346?start=675#overview
Course content
27. Packages 14min
28. Interfaces 19min
29. Public, Private, Protected 20min
30. Polymorphism 10min
31. Encapsulation and the API Docs 11min
32. Casting Numerical Values 11min
33. Upcasting and Downcasting 14min
34. Using Generics 13min
35. Generics and Wildcards 18min
Overview
Bookmarks
Announcements

About this course

Learn to program using the Java programming language

By the numbers | Skill level: All Levels | Lectures: 74


Java Tutorial for Complete B
udemy.com/course/java-tutorial/learn/lecture/152306#overview
Apps
Gmail
YouTube
Maps
Java - Using Generics/src/App.java - Eclipse
File Edit Run Source Refactor Navigate Search Project Window Help
*App.java
Writable
Smart Insert
Bookmarks
Announcements

```
import java.util.ArrayList;


public class App {

    public static void main(String[] args) {

        ////////////////// Before Java 5 //////////////////////
        ArrayList list = new ArrayList();

        list.add("apple");
        list.add("banana");
        list.add("orange");

        String fruit = (String)list.get(1);

        System.out.println(fruit);

        ////////////// Modern style //////////////////////////

        ArrayList<String> strings = new ArrayList<String>();

        strings.add("cat");
        strings.add("dog");
        strings.add("alligator");

        String animal = strings.get(1);

        System.out.println(animal);

        ///////////// There can be more than one type argument ////////////////////

        HashMap<Integer, String> map = new HashM
    }
```

About this course

Course content

34. Using Generics 13min
35. Generics and Wildcards 18min
36. Anonymous Classes 9min
37. Reading Files Using Scanne 13min
38. Handling Exceptions 16min
39. Multiple Exceptions 12min
40. Runtime vs. Checked Excep 9min
41. Abstract Classes 13min

toString() method:

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.

If you print any object, java compiler internally invokes the toString() method on the object. So overriding the toString() method, returns the desired output, it can be the state of an object etc. depends on your implementation.

Inheritance:

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:

- Encapsulation in Java is a mechanism of wrapping the data (variables) and code acting on the data (methods) together as a single unit.
- In encapsulation, the variables of a class will be hidden from other classes, and can be accessed only through the methods of their current class. Therefore, it is also known as **data hiding**.

Generic Methods

- All generic method declarations have a type parameter section delimited by angle brackets (< and >) that precedes the method's return type (< E > in the next example).
- Each type parameter section contains one or more type parameters separated by commas. A type parameter, also known as a type variable, is an identifier that specifies a generic type name.
- The type parameters can be used to declare the return type and act as placeholders for the types of the arguments passed to the generic method, which are known as actual type arguments.
- 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).

Java Type Casting

Type casting is when you assign a value of one primitive data type to another type.

In Java, there are two types of casting:

- **Widening Casting** (automatically) - converting a smaller type to a larger type size
  byte -> short -> char -> int -> long -> float -> double

- **Narrowing Casting** (manually) - converting a larger type to a smaller size type
  double -> float -> long -> int -> char -> short -> byte