

DAILY ASSESSMENT FORMAT

Date:	12/06/2020	Name:	Yashaswini.R
Course:	PCB design using Kicad	USN:	4AL17EC98
Topic:	1. Add footprint search path 2. Prepare production files	Semester & Section:	6th sem 'B' sec
Github Repository:	Yashaswini		

FORENOON SESSION DETAILS

Image of session

The screenshot displays the Udemy course interface for 'Learn KiCad. Printed Circuit Board Design.' The main video player shows a KiCad interface with a 'Footprint library files' dialog box open. The dialog box lists various footprint files like 'connect', 'discrete', 'pin_array', etc. The course content sidebar on the right shows a list of 8 lectures, with the 7th lecture 'Add Footprint search path' highlighted. The course title is 'Learn KiCad. Printed Circuit Board Design.' and the instructor is 'Udemy'.

Course content

Section 1: Up and running.
6 / 8 | 1hr 29min

- ☒ 1. Start a new project. 18min
- ☒ 2. Netlist and footprint association and placing PCB items. 16min
- ☒ 3. Silk-screen and copper pour. 9min
- ☒ 4. Mounting holes. 4min
- ☒ 5. Create a library and put your own component in that library. 9min
- ☒ 6. Create PCB footprint component. 12min
- ☒ 7. Add Footprint search path. 2min
- ☐ 8. Prepare production files. 20min

About this course

A guide to get you started to use the open source program KiCad for your next electronics project.

By the numbers Skill level: Beginner Level Lectures: 8

Learn KiCad. Printed Circuit

udemy.com/course/learn-kicad-printed-circuit-board-design/learn/lecture/1229108#overview

Apps Gmail YouTube Maps

Udemy

Learn KiCad. Printed Circuit Board Design.

★ Leave a rating

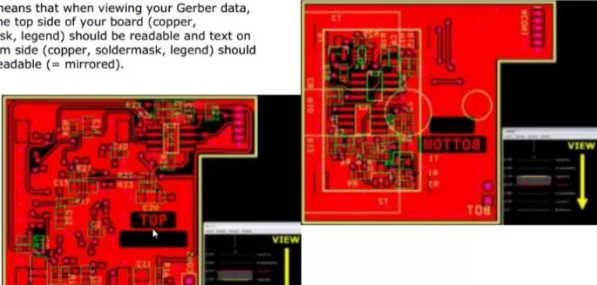
○ Your progress

Share

9. Make sure that your data is supplied as **seen from top to bottom through the PCB. DO NOT mirror** (or reflect) any data layer - image or drill.

Viewing a PCB from top to bottom through the board is the universal practice in the PCB industry. We view and handle your data in that way, as does your CAD PCB design software.

-> This means that when viewing your Gerber data, text on the top side of your board (copper, soldermask, legend) should be readable and text on the bottom side (copper, soldermask, legend) should be non-readable (= mirrored).



Overview

Notes

Announcements

About this course

A guide to get you started to use the open source program KiCad for your next electronics project.

By the numbers

Skill level: Beginner Level

Lectures: 8

Course content

Section 1: Up and running.
8 / 8 | 1hr 29min

✓ 1. Start a new project.
18min

✓ 2. Netlist and footprint association and placing PCB items.
16min

✓ 3. Silk-screen and copper pour.
9min

✓ 4. Mounting holes.
4min

✓ 5. Create a library and put your own component in that library.
9min

✓ 6. Create PCB footprint component.
12min

✓ 7. Add Footprint search path
2min

✓ 8. Prepare production files.
20min

Peel-off Mask Design Rule Specifications:

- Minimum width of any Peel-off element (P): 0.500mm (20mil)
- Maximum coverable hole ENDSIZE (H): 6.00mm (236mil)
- Minimum overlap on copper pattern (V): 0.600mm (24mil)
- Minimum clearance to free copper (W): 0.600mm (24mil)
- Minimum distance from PCB outline: 0.500mm (20mil)
- Tolerance on position: +/-0.300mm (12mil)

Clipping Rules:

- Legend clipping clearance is 0.100mm (4mil) back from the Soldermask openings.
- Any parts of line smaller than 0.100mm (4mil) will be removed.

Board Outline:

- Please always include the board outline in your legend layers output data as this helps avoid any potential misalignment, mirroring or rotation issues.
- This is best done using a small line – e.g. 0.500mm (20mil) wide – where the center of the line is the exact board outline.
- We will remove this line in the actual production data we use for manufacturing your boards.
- In all cases we will clip away any legend text within 0.200mm (10mil) of the board edge.

DAILY ASSESSMENT FORMAT

Date:	12/06/2020	Name:	Yashaswini.R
Course:	Java	USN:	4AL17EC098
Topic:	Programming core java	Semester & Section:	6 th sem 'B' sec
Github Repository:	Yashaswini		

AFTERNOON SESSION DETAILS

Image of session

The screenshot displays the Udemy interface for the course 'Java Tutorial for Complete Beginners'. The main video player shows a Java IDE with the following code:

```
public static void main(String[] args) {  
    ArrayList<Machine> list1 = new ArrayList<Machine>();  
    list1.add(new Machine());  
    list1.add(new Machine());  
    ArrayList<Camera> list2 = new ArrayList<Camera>();  
    list2.add(new Camera());  
    list2.add(new Camera());  
    show(list1);  
}  
  
public static void show(ArrayList<Machine> list1) {  
    for (Machine value : list1) {  
        System.out.println(value);  
    }  
}
```

The course content list on the right includes the following topics:

- 35. Generics and Wildcards (18min)
- 36. Anonymous Classes (9min)
- 37. Reading Files Using Scanner (13min)
- 38. Handling Exceptions (16min)
- 39. Multiple Exceptions (12min)
- 40. Runtime vs. Checked Exceptions (9min)
- 41. Abstract Classes (13min)
- 42. Reading Files With File Reader (17min)
- 43. Try-With-Resources (11min)

The 'About this course' section states: 'Learn to program using the Java programming language'. The course has 74 lectures and is suitable for all skill levels.

The screenshot displays the UdeMy interface for the course 'Java Tutorial for Complete Beginners'. The main video player shows a Java code snippet for a class named 'App' with a 'main' method. The course content list on the right includes topics like 'Abstract Classes', 'Reading Files With File Reader', 'Try-With-Resources', 'Creating and Writing Text Files', 'The Equals Method', 'Inner Classes', 'Enum Types: Basic and Advanced Usage', 'Recursion: A Useful Trick Up Your Sleeve', 'Serialization: Saving Objects to Files', and 'Serializing Arrays'. The 'About this course' section at the bottom states: 'Learn to program using the Java programming language', 'By the numbers', 'Skill level: All Levels', and 'Lectures: 74'.

Generics and Wildcards:

Java Generic's wildcards is a mechanism in Java Generics aimed at making it possible to cast a collection of a certain class, e.g A, to a collection of a subclass or superclass of A

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).

Anonymous Class :

- It is an inner class without a name and for which only a single object is created.
- An anonymous inner class can be useful when making an instance of an object with certain “extras” such as overloading methods of a class or interface, without having to actually subclass a class.
- Anonymous inner classes are useful in writing implementation classes for listener interfaces in graphics programming.

Reading Files Using Scanner:

- A simple text scanner which can parse primitive types and strings using regular expressions.
- A Scanner breaks its input into tokens using a delimiter pattern, which by default matches whitespace. The resulting tokens may then be converted into values of different types using the various methods.

Exception Handling:

If an exception occurs, which has not been handled by programmer then program execution gets terminated and a system generated error message is shown to the user.

Abstract classes:

Abstract classes allow you to define the parent class of a new hierarchy without having to worry about the user actually instantiating the parent