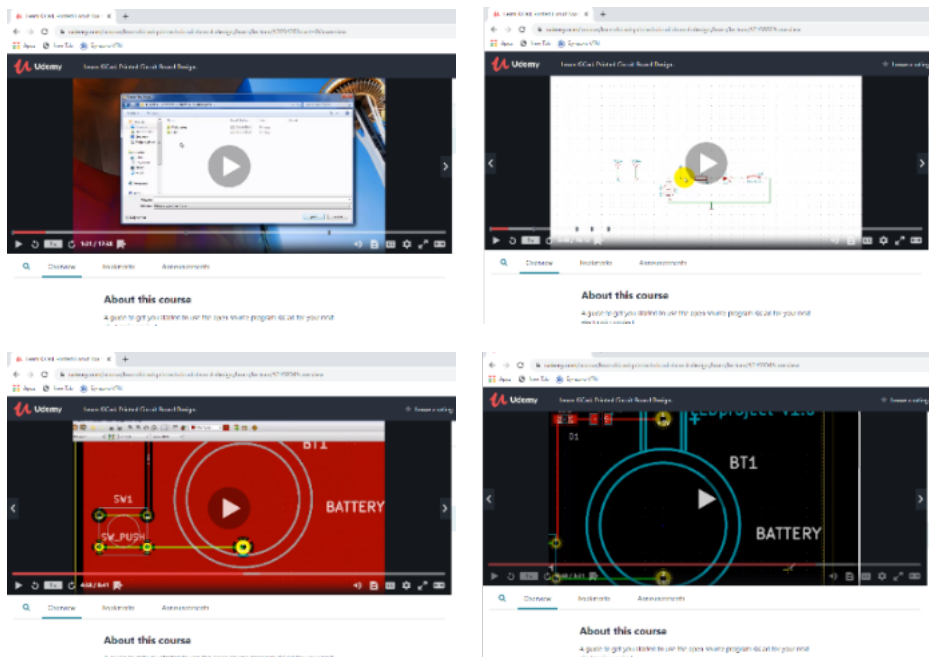


Date:	09-06-2020	Name:	Yamunashree N
Course:	PCB	USN:	4AL17EC097
Topic:	Start a new project,netlist and footprint generator.	Semester and section:	6 <sup>th</sup> sem and 'B' sec
Repository name:	yamunashree-course		



## FOOTPRINTS PLACEMENT

Our project is now at the stage where the two footprints that compose our PCB are spread out in the Pcbnew canvas. In this chapter we will do the footprint placement so that we can start giving shape to the final PCB. I would like to place the connector on the right side of the breakout and the nRF24 component on the left side. To do this, position the cursor over the nRF24 footprint and hit the 'M' key. This will allow you to move this footprint. Move it so that it is on the left side of the straight connector. The nRF24 footprint is on the left side of the connector. Notice the rattiest lines. They indicate the pads that should be wired together. The thin white lines that connect the pads together are called "ratsnests". They are routing guides. As we wire each pair of pads together, the corresponding ratsnest will disappear. One thing to consider when you are placing your components onto your PCB is space. How much space is your final PCB going to take up? Remember that PCB manufacturers will charge you not based on how many holes and tracks your PCB has but based on its dimensions. Therefore, the smaller your PCB is, the cheaper it will be to make.



However, the smaller the PCB is, the harder can be to route it. With less space in between footprints, the routing of tracks will be more difficult. This is not a problem for the simple PCB of this example, however if you had more footprints, then placing them too close to each other would make routing and then soldering harder. You must think about this and find a dimension that works both from a cost point of view and from these other technical considerations like the soldering and the routing point of view.

### Silkscreen and copper pour

Unfortunately, when part outlines and text are drawn on the copper layer, they are electrically conductive. You can't put traces or parts in the same places, as the copper text would interfere and change the circuit. In your layout program, you can place text and part outlines on the silkscreen layer instead of copper. For example, in Copper Connection simply choose the silkscreen layer before placing text, or select the existing text and switch the layer to silkscreen. Copper Connection includes a bulk selection feature for your convenience. If you have already made a board with text on a copper layer, you don't have to move each text element one at a time. Instead, right click on any copper-layer text, choose "Select All Text on this Layer", and then choose the new layer (top or bottom silkscreen). Now, when you order the boards from your favorite manufacturer, be sure to choose one of their manufacturing options that includes silkscreen. For prototype runs, silkscreen is usually restricted to the top side of the board, but some manufacturers offer both sides.



Date:	09-06-2020	Name:	Yamunashree N
Course:	JAVA	USN:	4AL17EC097
Topic:	Programming CORE JAVA	Semester and section:	6 <sup>th</sup> sem and B sec

classmate  
Date \_\_\_\_\_  
Page \_\_\_\_\_

# JAVA

## # Programming core java

- file → new java project → give name → finish
- Right Click on project folder → new → class  
windows will popup  
↓  
give the class name → Method stubs like to create  
↓  
finish

```

public class Application {
    public static void main (String[] args) {
        System.out.println("Hello world");
    }
}

```

- Using Variable :-  

```

public class Application {
    public static void main (String[] args) {
        int myNumber = 88;
        short myShort = 847;
        long myLong = 9797;
        double myDouble = 7.3243;
        float myFloat = 324.3f;
        char myChar = 'Y';
        boolean myBoolean = false;
        byte myByte = 127;
        System.out.println(myNumber);
        System.out.println(myShort);
        System.out.println(myByte);
    }
}

```

• Strings : Working with text : —

```
public class Application {
    public static void main (String[] args) {
        int myInt = 7;
        String text = "Hello";
        String blank = " ";
        String name = "john";
        String greeting = text + blank + name;
        System.out.println(greeting);
        System.out.println("Hello" + " " + "John");
        System.out.println("My integer is : " + myInt);
        double myDouble = 7.8;
        System.out.println("My number is : " + myDouble + ".");
    }
}
```

• While loops : —

```
public class Application {
    public static void main (String[] args) {
        int value = 0;
        while (value < 10)
        {
            System.out.println("Hello" + value);
            value = value + 1;
        }
    }
}
```

• For loops :-

```
for (int i = 0; i < 5; i++) {
    System.out.println("The value of i is : " + i);
}
```



• If :-

```
int loop = 0;  
while (true) {  
    System.out.println("looping: " + loop);  
    if (loop == 5) {  
        break;  
    }  
    loop++;  
    System.out.println("Running");  
}
```