**DAILY ASSESSMENT FORMAT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **11/06/2020** | **Name:** | **Lavanya B** |
| **Course:** | **Kicad** | **USN:** | **4al17ec043** |
| **Topic:** | **Design principles and basic concepts**  **Additional design consideration** | **Semester & Section:** | **6th A** |
| **Github Repository:** | **Lavanya-B** |  |  |

|  |
| --- |
| **FORENOON SESSION DETAILS** |
| **Image of session** |
| **Report**  **Design principles and basic concepts**  **Add footprint search path**    **In KiCad, one can define some paths using an environment variable. A few environment variables are internally defined by KiCad, and can be used to define paths.**  **This is useful when absolute paths are not known or are subject to change. This is the case for "official" libraries built for KiCad:**  **-for the path of these libraries, when installed on your disk**  **-for the path of 3D shapes files used in footprint definitions.**  **For instance, the full path of connect.pretty footprint library is defined like this, when using the KISYSMOD environment variable to define the full path: ${KISYSMOD}/connect.pretty**  **Obviously, one can use a usual full path definition, if this full path is well known, and never changes.**  **Gerber file**  **The Gerber format is an open ASCII vector format for printed circuit board (PCB) designs. It is the de facto standard used by PCB industry software to describe the printed circuit board images: copper layers, solder mask, legend, drill data.**  **The file composed of 4 components, they are**   1. **Configuration parameters** 2. **Aperture definitions** 3. **XY coordinate locations for draw and flash commands** 4. **Draw and flash command codes**   **Drill file**  **In any PCB, there are typically many holes that need to be drilled for things such as through-hole parts, vias and mounting holes. There are multiple files relating to the drilling of the holes in the PCB. The one essential in the creation of the PCB is the NC Drill file. The drill files can be generated by nearly all PCB design software with its ordinary format as Excellon that was originally a manufacturer of CNC systems for drillers and routers. Owing to its long life and active role in PCB drilling and routing equipment, Excellon has been accepted and coined as a set of industry standards.**  **Project file and manufacturing file**  **The file format for PCB manufacturing is called Gerber. One Gerber file provides information for one PCB feature on one layer. Thus, if you have a two-layer board and each side has copper, solder mask, and silkscreen, you will need six Gerber files. If you’d prefer to avoid generating any type of manufacturing file, you can look for a PCB**  **manufacturer that accepts your CAD software’s project files. I assume that the**  **manufacturer uses some sort of automated procedure to generate Gerbers from the project**  **file. This is beneficial not only because it saves you time but also because the fab house**  **technicians will know exactly how to generate files that are compatible with**  **their equipment. The listin the next section gives some information about one manufacturer**  **that accepts project files.** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date:** | **11/06/2020** | **Name:** | **Lavanya B** | |
| **Course:** | **JAVA** | **USN:** | **4al17ec043** | |
| **Topic:** |  | **Semester & Section:** | **6th A** | |
| **AFTERNOON SESSION DETAILS** | | | |
| **Image of session** | | | |
| **Report**  **Programming core JAVA**  **Static**  **When we declare a variable or a method as static, it belongs to the class, rather than to a specific instance. This means that only one instance of a static member exists, even if you create multiple objects of the class, or if you don't create any. It will be shared by all objects.**  **Eg:**  **public class Vehicle {**  **public static void clr() {**  **System.out.println("Black");**  **}**  **}**  **public class MyClass {**  **public static void main(String[ ] args) {**  **Vehicle.clr();**  **}**  **}**  **Package**  **Packages are used to avoid name conflicts and to control access to classes.**  **A package can be defined as a group made up of similar types of classes, along with sub-packages.**  **Creating a package in Java is quite easy. Simply right click on your src directory and click New->Package. Give your package a name and click Finish.**  **import samples.Vehicle;**  **class MyClass {**  **public static void main(String[ ] args) {**  **Vehicle v1 = new Vehicle();**  **v1.horn();**  **}**  **}**  **Where sample is a package and Vehicle is a class name.**  **Encapsulation**  **Encapsulation is to ensure that implementation details are not visible to users. The variables of one class will be hidden from the other classes, accessible only through the methods of the current class. This is called data hiding.**  **To achieve encapsulation in Java, declare the class' variables as private and provide public setter and getter methods to modify and view the variables' values.**   * **Control of the way data is accessed or modified** * **More flexible and easily changed code** * **Ability to change one part of the code without affecting other parts**   **Inheritance** | | | |