**DAILY ASSESSMENT FORMAT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **12/06/2020** | **Name:** | **Namratha S Hipparagi** |
| **Course:** | **VLSI design** | **USN:** | **4AL16EC040** |
| **Topic:** | **MOS transistor basics** | **Semester & Section:** | **8 A** |
| **Github Repository:** | **namrathahipparagi\_1** |  |  |

|  |
| --- |
| **FORENOON SESSION DETAILS** |
| **Image of session** |
| **Report**  **MOSFET**  The metal–oxide–semiconductor field-effect transistor, also known as the metal–oxide–silicon transistor, is a type of insulated-gate field-effect transistor that is fabricated by the controlled oxidation of a semiconductor, typically silicon. Construction The construction of a MOSFET is a bit similar to the FET. An oxide layer is deposited on the substrate to which the gate terminal is connected. This oxide layer acts as an insulator (sio2 insulates from the substrate), and hence the MOSFET has another name as IGFET. In the construction of MOSFET, a lightly doped substrate, is diffused with a heavily doped region. Depending upon the substrate used, they are called as P-type and N-type MOSFETs. The voltage at gate controls the operation of the MOSFET. In this case, both positive and negative voltages can be applied on the gate as it is insulated from the channel. With negative gate bias voltage, it acts as depletion MOSFET while with positive gate bias voltage it acts as an Enhancement MOSFET. The N-channel MOSFETs are simply called as NMOS.  The following figure shows the construction of a MOSFET.  MOSFET Construction  N-Channel MOSFET  The **P-channel MOSFETs** are simply called as **PMOS**.  P-channel MOSFET |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date:** | **12/6/2020** | **Name:** | **Namratha S Hipparagi** | |
| **Course:** | **Java** | **USN:** | **4al16ec040** | |
| **Topic:** | **Generics and Wildcards, Anonymous Classes, Reading Files Using Scanner, Handling Exceptions, Multiple Exceptions….** | **Semester & Section:** | **8 A** | |
| **AFTERNOON SESSION DETAILS** | | | |
| **REPORT** **Wildcards in Java** It is known as the wildcard in generic programming. It represents an unknown type. The wildcard can be used in a variety of situations such as the type of a parameter, field, or local variable; sometimes as a return type. The wildcard is an advanced search technique that can be used to maximize your search results in library databases. Wildcards are used in search terms to represent one or more other characters. The two most commonly used wildcards are: An asterisk (\*) may be used to specify any number of characters. Unlike arrays, different instantiations of a generic type are not compatible with each other, not even explicitly. **Abstract Classes in Java** A class which is declared as abstract is known as an abstract class. It can have abstract and non-abstract methods. It needs to be extended and its method implemented. It cannot be instantiated. In C++, if a class has at least one pure virtual function, then the class becomes abstract. Unlike C++, in Java, a separate keyword abstract is used to make a class abstract.  // An example abstract class in Java  abstract class Shape {  int color;  // An abstract function (like a pure virtual function in C++)  abstract void draw();  }  //Java program to demonstrate Upper Bounded Wildcards  import java.util.Arrays;  import java.util.List;  class WildcardDemo  {  public static void main(String[] args)  {    //Upper Bounded Integer List  List<Integer> list1= Arrays.asList(4,5,6,7);    //printing the sum of elements in list  System.out.println("Total sum is:"+sum(list1));  //Double list  List<Double> list2=Arrays.asList(4.1,5.1,6.1);    //printing the sum of elements in list  System.out.print("Total sum is:"+sum(list2));  }  private static double sum(List<? extends Number> list)  {  double sum=0.0;  for (Number i: list)  {  sum+=i.doubleValue();  }  return sum;  }  } | | | |