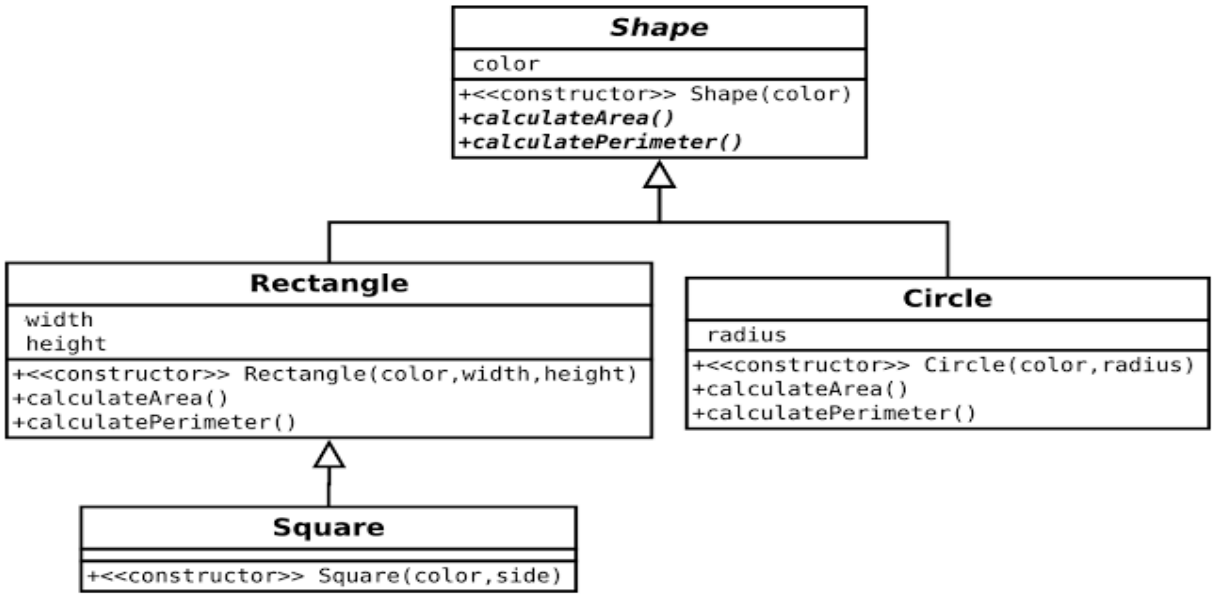
Lesson – 5 – Inheritance and Polymorphism.

Note : Need to solve all the problems in a Polymorphic way.

Day – 1

Problem : 1

Look at the UML & design the four different classes such as Shape.java, Rectengle.java, Circle.java and Square.java as per the relationship of the given diagram.



Perform the following tasks

a. The return value of calcualteArea() & calculatePerimeter() is double.
b. Don't provide any getter or setters, make all the fields as either default or protected.
c. In the Shape class just return 0.0 for the calcualteArea() & calculatePerimeter().
d. Square class has no attributes. Assign the side value to height and width attribute.
e. Create a main class to test the application to perform the below tasks
    a. Create an array of Shape[] to hold at-least five objects for its subtypes.
    b. Create a Static method to print the total area of all the Shape objects and print the total perimeter of all the Shape objects.

```
public static void printTotal(Shape[] shapes)
  { // Implement your code
    }
```

Area of Rectangle = width * height	Perimeter of Rectangle = 2 * width + 2 * height
Area of Circle = π * r * r	Area of Circle = 2 * π * r
Area of Square = side * side	Area of Square = 4 * side

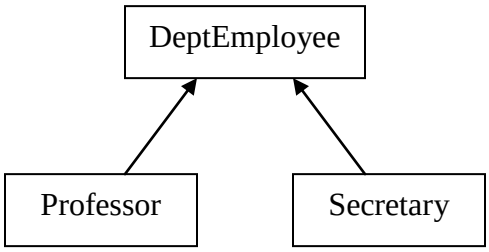
Problem : 2

A university department consists of professors and secretaries. Each professor and each secretary has a name, a salary, and a hire date. Use inheritance and polymorphism to create an application that represents the department and its professors and secretaries as objects, and provides a test class that creates 3 professors and 2 secretaries, and then outputs the combined total of all of their salaries.

Start by creating classes

Professor
Secretary
DeptEmployee

having the following relationship:



Place instance fields and corresponding accessor/mutator methods in DeptEmployee to represent name and hire date (as a type of LocalDate)(do not create accessors or mutators for salary). Do not put these fields in either the Professor or Secretary class. Also place in the Professor class an int field numberOfPublications, with corresponding accessor and mutator methods. Place in the Secretary class a double field overtimeHours, also with corresponding accessor and mutator methods.

Place a computeSalary method in DeptEmployee which simply returns the value stored in salary. Override the computeSalary method in Secretary so that the return value is the sum of the value of salary *plus* 12 times the number of overtime hours.

Then in the main method of a class named Main, create three instances of Professor and two instances of Secretary (you can invent the values to pass into the constructor). Finally, create an array of department employees:

```
DeptEmployee[] department = new DeptEmployee[5]
```

and then populate the array with the Professor and Secretary instances you have just created. Then ask the user if he wishes to see the sum of all Professor salary, sum of all secretary salary and all salaries in the department. If the user responds "Y", then loop through the department array and polymorphically read, and sum, all salaries, and output the result to the console.

Day – 2

Problem : 3

Here is a list of the things you need to do for this problem:

a. Create an abstract class Figure with void getFigure() method need to print the figure.
b. Create sub classes for the Figure class such as UpwardHat, DownwardHat, FaceMaker and Vertical. You have to override the getFigure() method to display the figures /\, \/,:), and ||.
c. Create your main method for the Driver class. Apply polymorphism to create an array of objects for Figure class. The output after running the main method should look like this.

   /\ /\ \/ :) ||

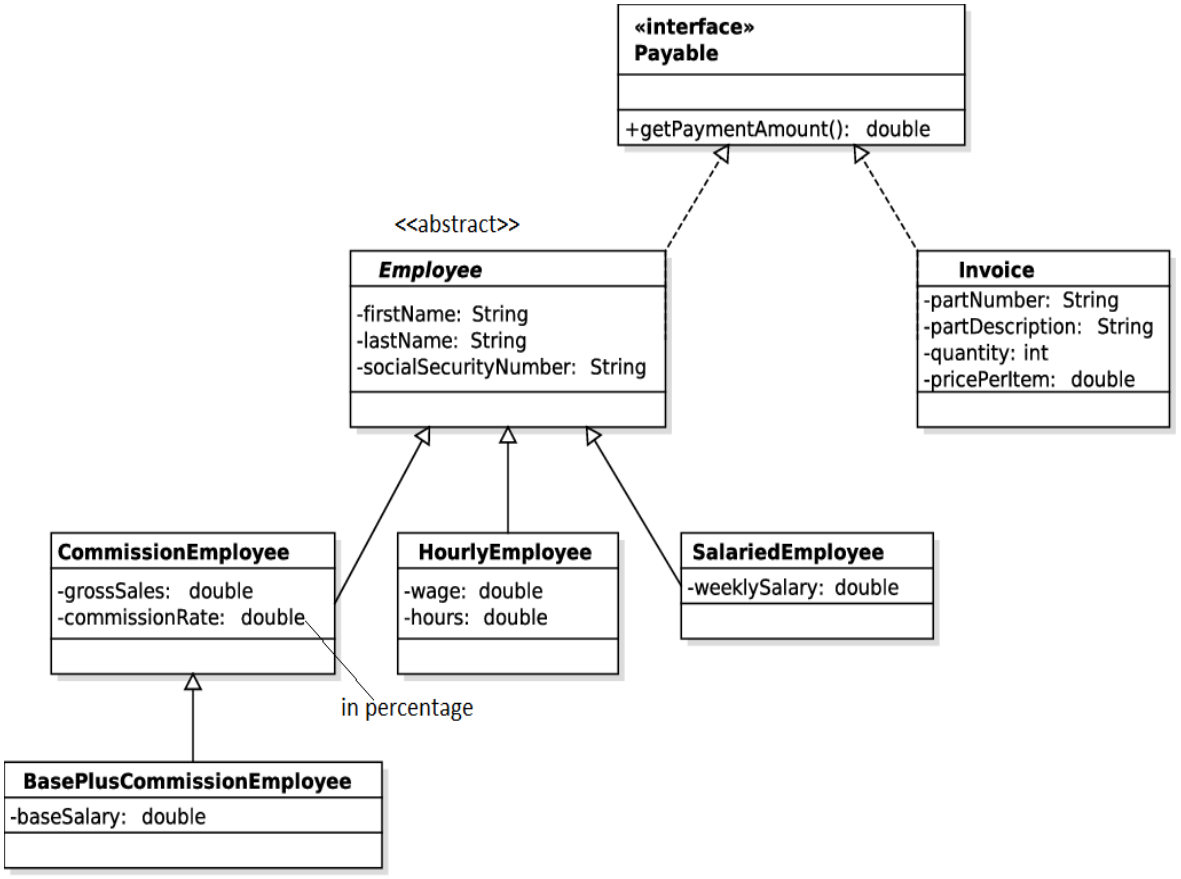
d. Again print the same output along with their class name as look like this.

   UpwardHat : /\
   UpwardHat : /\
   DownwardHat : \/
   FaceMaker : :)
   Vertical : | |

Problem - 4

1. Write a code for the given UML Diagram.

1. Provide necessary getters and setters
2. Provide necessary constructors to initialize values
3. Override the toString() method to display the current status of the objects
4. Write a driver class to test by creating an array of five objects for various employee categories and one object for Invoice class.
5. Calculate the total salaries of all the employees and print it into the console.



Hints : The getPaymentAmount() need to return the values as mentioned below according to the specific class object.

1. CommissionEmployee : grossSales * CommisionRate
2. BasePlusCommisionEmployee : baseSalary + (grossSales * CommisionRate)
3. HourlyEmployee : wage * hours
4. SalariedEmployee : weeklySalary
5. Invoice : quantity * pricePerItem

Day – 3

Problem : 5

Implement the Java code for the given class diagram.

Computer
String manufacturer String processor int ramSize int diskSize double processorSpeed

```
// Make a Constructor to initialize the instance fields
int getRamSize()
int getDiskSize()
double getProcessorSpeed()
double computePower() // return ramSize multiplied by processorSpeed
Override the toString() method to display the current status
```

a) Override the equals() and hashCode() method for the Computer class.

b) Write a Main class to create an object for Computer to test the operations.

c) Compare two Computer Objects are equal or not.

Problem : 6

Create a class called Person.

```
class Person
  {
    String name;
    Computer computer; // Refer from the Problem 3.
  }
```

a. Apply clone() method and write the implementation to understand the concepts of shallow copy.

b. Apply clone() method and write the implementation to understand the concepts of deep copy.