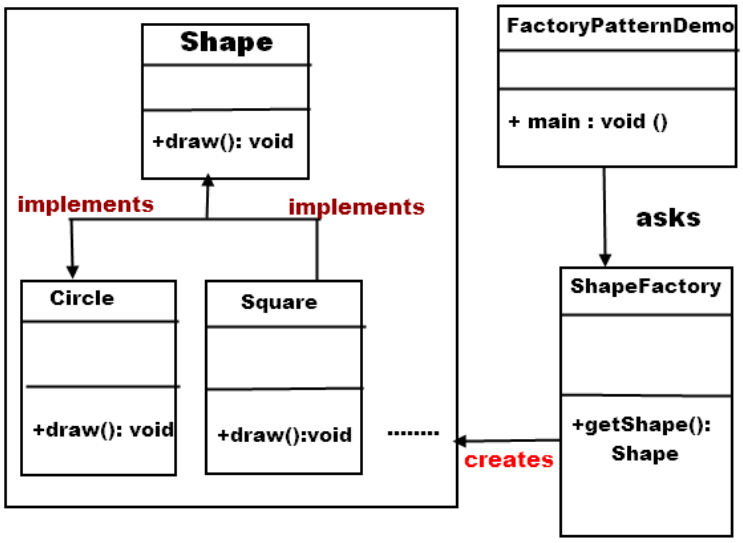
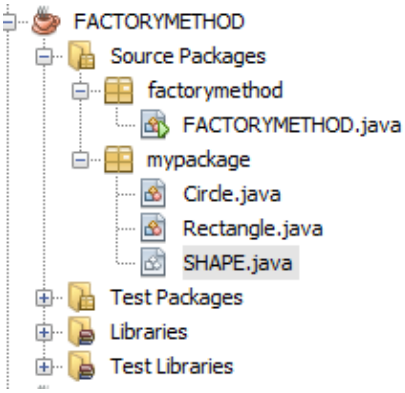
JAVA ASSIGNMENT 2
Paper code: MCAN 293

Purpose: To know the use of inheritance concepts, code reusability, and anonymous class concepts.

1. Implement the following UML class Diagram where Shape is an interface where you will declare the draw function. From the main function user request for a specific shape and shape factory will create that shape according to the request.



To implement the above problem, you should follow the following class hierarchy



The factory method code is given below:

```
public class FACTORYMETHOD {
   public static SHAPE getShape(String shapetype){
```

```
    // Complete the code here
}

public static void main(String[] args) {
    String shapetype;
    Scanner sc = new Scanner(System.in);
    System.out.println("Enter the shape ");
    shapetype = sc.nextLine();

    SHAPE object = getShape(shapetype);
    object.draw();

}//END OF MAIN
}//END OF CLASS
```

2. Create a base class called Vehicle that stores of wheels and speed. Create the derived classes:

   - Car that inherits Vehicle and also stores number of passengers.
   - Truck that inherits Vehicle and also stores the load limit.

   Write a main() function to create objects of these classes and display all the information about the car and truck. Then compare the speed of the two vehicles (car and truck), and return the proper vehicle whose speed is larger than another one. It should display a "faster "or "slower" message if the car is faster or slower than the truck.

3. There are two classes named college and student. Each student has roll, name, stream, admission year and marks. Each college has an id and college name. College has many students but a student can register his/her name only in one college. Write an application that creates the student object and add them to the college. Print the number of students in a college. Print all the student information of a specific college.

4. Suppose you have an interface name IOne. The interface has a function f1() with return type void. Implement the interface using anonymous class concepts.