Day-4 Assignment Questions:

1. We want to store the information about different vehicles. Create a class named Vehicle with two data member named mileage and price. Create its two subclasses
---Car with data members to store ownership cost, warranty (by years), seating capacity and fuel type (diesel or petrol).
---Bike with data members to store the number of cylinders, number of gears, cooling type(air, liquid or oil), wheel type(alloys or spokes) and fuel tank size(in inches)
Make another two subclasses Audi and Ford of Car, each having a data member to store the model type.
Next, make two subclasses Bajaj and TVS, each having a data member to store the make-type.
Now, store and print the information of an Audi and a Ford car (i.e. model type, ownership cost, warranty, seating capacity, fuel type, mileage and price.) Do the same for a Bajaj and a TVS bike.

2. Construct a base class called twoD contains x and y and methods to read and write the x and y.
Create another class called threeD which is derived from twoD and also contains z and write methods to read and write z.
Also write a method to find the distance of two threeD Points.
sqrt((x2-x1)^2+(y2-y1)^2+(z2-z1)^2)
In main:
Create one ThreeD object using the default constructor.
Use the setters to set x, y, and z.
Create the second ThreeD object using the constructor with three arguments.
in the TwoD class:
Add a cout statement to the TwoD default constructor with a message "TwoD default constructor"
Add a cout statement to other TwoD constructor with a message "TwoD constructor with two arguments"

3. A class Point is declared as follows:

```
class Point{
public:
Point(){int=0;int=0;} //default constructor
void setPoint(int,int); //set coordinates
int getX() const {return x;} //get x coordinates
int getY() const {return y;} //get y coordinates
void printPoint(); // print (x,y) coordinates
private int x;
private int y;
};
```

Write the implementation of the class Point in the same file.
Then, declare a class called Circle that is derived from the class Point.
The class Circle has public member functions Circle (constructor), setRadius(), getRadius() and area()
and one private data member radius. The class Circle indirectly uses private member x and y of class Point to store the coordinate of the center of the circle.
The class Circle also checks to make sure the radius value is a positive number, otherwise it is set to default value 1.
Note: The private members of class Point can only be indirectly accessed by class Circle using public methods of class point.
Implement the class Circle and write a driver program to test the class Circle.
An example of the output of the driver program is.

```
Enter x: 2
Enter y: 2
Enter radius: 1
Circle center is (2,2)
Radius is 1
Area is 3.14
```

4. Write a program to illustrate that Classes cannot be used for multiple Inheritance.

5. Create a class named Shape with a method that prints "This is a shape". Create another class named
Polygon inheriting the Shape class with the same method that prints "Polygon is a shape".
Create two other classes named Rectangle and Triangle having the same method which prints "Rectangle is a polygon"
and "Triangle is a polygon" respectively. Again, make another class named Square having the same method which prints
"Square is a rectangle".
Now, try calling the method by the object of each of these classes.

6. Design and implement a custom Java class named MyString that mimics the behavior of Java's built-in String class.
Your class should store character data internally
(e.g., using a char[] or String as input) and should provide the following string methods,
1. int length() – Returns the number of characters.
2. char charAt(int index) – Returns the character at the specified index.
3. boolean equals(MyString other) – Checks if two MyString objects are equal.
4. MyString toUpperCase() – Returns a new string with all characters in uppercase.
5. MyString toLowerCase() – Returns a new string with all characters in lowercase.
6. MyString substring(int start, int end) – Returns a substring from start to end-1.
7. MyString concat(MyString other) – Concatenates another string to the current one.
8. boolean contains(MyString sub) – Checks if a substring exists.
9. int indexOf(char ch) – Returns the index of the first occurrence of the character.
10. MyString replace(char oldChar, char newChar) – Replaces all occurrences of a character.
* Implement proper constructors, including one that takes a char[] or String as input.
* Avoid using any built-in String methods to perform the operations internally.
* Write a main method to demonstrate the working of your MyString class.

7. Design a calculator application using Java Inheritance.
Create the following class hierarchy:

--->BasicCalculator (Base class):
Implement at least three basic arithmetic methods, such as:

--->add(int a, int b)
--->subtract(int a, int b)
--->multiply(int a, int b)
--->divide(int a, int b)

--->AdvancedCalculator (Inherits from BasicCalculator):
Add 3 to 4 advanced mathematical operations, such as:

--->power(int base, int exponent)
--->modulus(int a, int b)
--->squareRoot(double number)

--->ScientificCalculator (Inherits from AdvancedCalculator):
Add scientific functions, such as:

--->sin(double angle)

--->cos(double angle)
--->log(double value)
--->exp(double value)
* Demonstrate the use of inheritance by creating an object of ScientificCalculator and calling methods from all three levels of the class hierarchy.
* Use appropriate access specifiers and method overrides where required.
* Add a main() method to test all operations.