1) Create a class that keeps track of the number of instances created. Implement a static variable and method to accomplish this

Ans:

```
public class InstanceCounter {
    private static int instanceCount = 0;

    public InstanceCounter() {
        instanceCount++;
    }

    public static int getInstanceCount() {
        return instanceCount;
    }
}

public class Main {
    public static void main(String[] args) {
        InstanceCounter obj1 = new InstanceCounter();
        InstanceCounter obj2 = new InstanceCounter();
        InstanceCounter obj3 = new InstanceCounter();

        int count = InstanceCounter.getInstanceCount();
        System.out.println("Number of instances created: " + count);
    }
}
```

Output: Number of instances created: 3

2) Write a program to create a constructor with parameters and initialise the variable using a constructor

Ans:

```
public class Student {
    private String name;
    private int age;

    public Student(String name, int age) {
```

```
        this.name = name;
        this.age = age;
    }

    public void displayInfo() {
        System.out.println("Name: " + name);
        System.out.println("Age: " + age);
    }

    public static void main(String[] args) {
        Student student1 = new Student("John", 20);
        student1.displayInfo();

        Student student2 = new Student("Jane", 22);
        student2.displayInfo();
    }
}
```

Output :

```
 Name: John
Age: 20
Name: Jane
Age: 22
```

3) Use a private keyword for a variable and use setter and getter method to initialize and print the values

Ans:

```
public class Student {
  private String name;

  private int age;

  public void setName(String name) {

            this.name = name;
```

```
}
    public void setAge(int age) {
        this.age = age;
    }
    public String getName() {
        return name;
    }
    public int getAge() {
        return age;
    }
    public static void main(String[] args) {
        Student student1 = new Student();
        student1.setName("John");
        student1.setAge(20);

        System.out.println("Name: " + student1.getName());
        System.out.println("Age: " + student1.getAge());
```

```
        Student student2 = new Student();

        student2.setName("Jane");

        student2.setAge(22);



        System.out.println("Name: " + student2.getName());

        System.out.println("Age: " + student2.getAge());

    }

}
```

Output

```
Name: John
Age: 20
Name: Jane
Age: 22
```

4) Write a program to call a method without creating an object of a class

Ans:

```
public class MethodWithoutObject {

    public static void displayMessage() {
        System.out.println("Display message!");
    }

    public static void main(String[] args) {
        displayMessage();
    }
}
```

5) Write a program which has static block and constructor overloading, initialize variables using the constructor and print it

Ans:

```
public class InitializationExample {

    private static int staticVariable;
    private int instanceVariable;

    static {
        // Static block
        staticVariable = 10;
        System.out.println("Static block executed");
    }

    public InitializationExample() {
        // Default constructor
        instanceVariable = 20;
    }

    public InitializationExample(int value) {
        // Parameterized constructor
        instanceVariable = value;
    }

    public void displayValues() {
        System.out.println("Static variable: " + staticVariable);
        System.out.println("Instance variable: " + instanceVariable);
    }



    public static void main(String[] args) {


        InitializationExample obj1 = new InitializationExample();
        obj1.displayValues();
```

```
        InitializationExample obj2 = new InitializationExample(30);
        obj2.displayValues();
    }
}
```