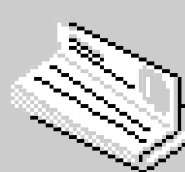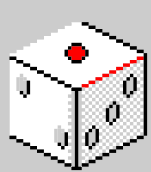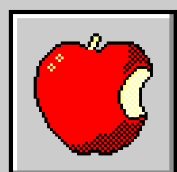# INNER CLASS

SAI VARSHINI BISANA

11:11PM

Contents

Start

Agenda

- <u>What is Inner Class</u>

- <u>Advantages of Inner class</u>

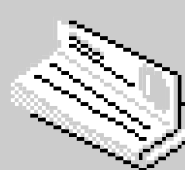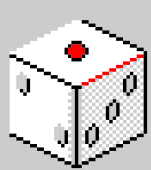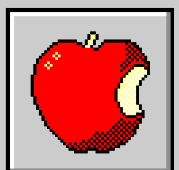- <u>Types of Inner class</u>
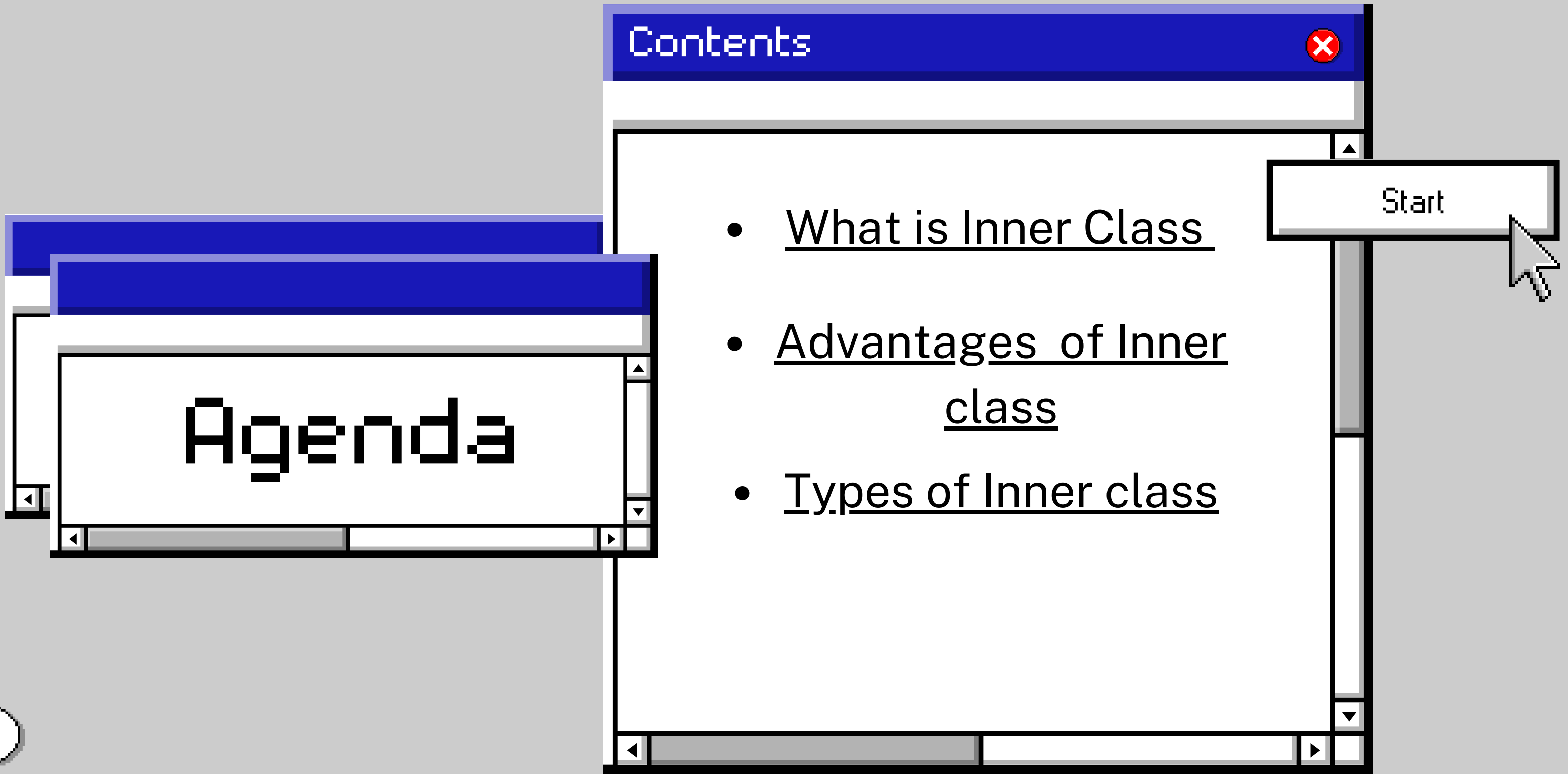
Introduction

- Inner class is a class declared inside a class or inteface.
- It is also known as Nested class.
- Inner classes have access to the members, including private members, of the outer class

```
Syntax of Inner class
class Java_Outer_Class{
            //code
class Java_Inner_Class{
            //code
            }
            }
```
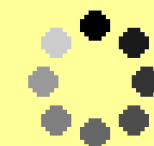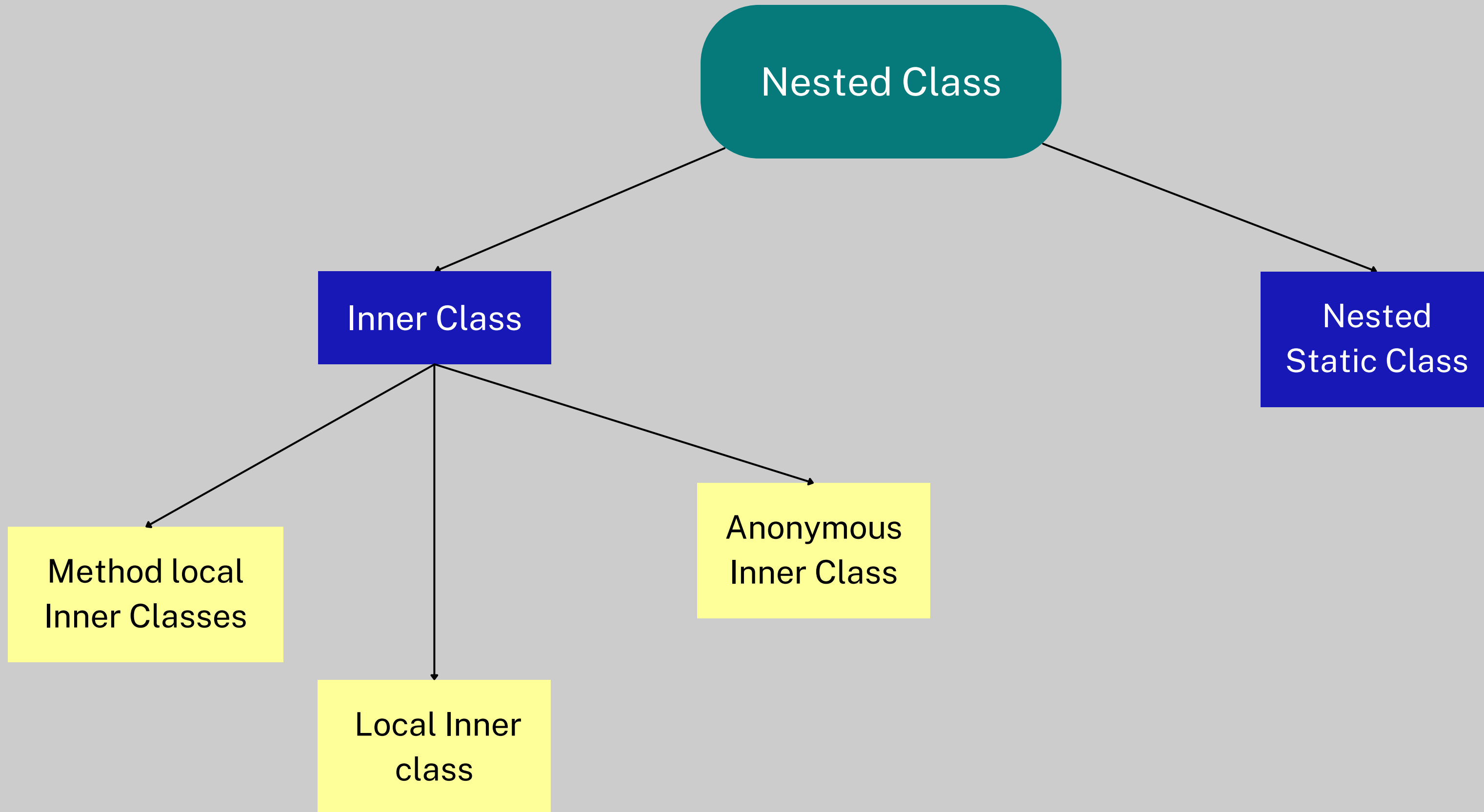
Advantages

- **Code optimization**
- **To develop a more readable and maintainable code**
- **Encapsulation and Organization**
- **Access to Outer Class Members**

Back to Agenda Page

Types of Inner class

Nested Class

Inner Class

Nested Static Class

Method local Inner Classes

Local Inner class

Anonymous Inner Class

Local Inner Class	A Local inner class is defined at the member level of a class
Metod Local Inner Class	A local inner class is defined within a block of code, typically within a method.
Anonymous Inner Class	An anonymous inner class is a class that is not given a name. It is created by implementing an interface or extending a class.
Static Nested Class	A static nested class is a class declared as static within the body of another class.

Local Inner Class

```java
public class OuterClass {
  private int outerField;

  public class InnerClass {
    public void display() {
      System.out.println("OuterField: " + outerField);
    }
  }
}
```

Method Local Inner Class

```java
public class OuterClass {
  public void outerMethod() {
    final int localVar = 10;

    class LocalInnerClass {
      public void display() {
        System.out.println("LocalVar: " + localVar);
      }
    }

    LocalInnerClass inner = new LocalInnerClass();
    inner.display();
  }
}
```

Anonymous Inner Class

```java
public class OuterClass {
  public void displayMessage() {
    OuterClass outer = new OuterClass(){
      @Override
      public void showMessage() {
        System.out.println("Hello from anonymous inner class!");
      }
    };

    myInterface.showMessage();
  }
}
}
```

```java
public class OuterClass {
    private static int outerStaticField;

    public static class NestedClass {
        public void display() {
            System.out.println("OuterStaticField: " + outerStaticField);
        }
    }
}
```

Thank you!