Unit 2 code

- **Classes and Objects**
- **Constructors**
- **Method Overloading**
- **Static Keyword**
- **`this` Keyword**
- **Garbage Collection**
- **Finalize Method**
- **Nested and Inner Classes**

1. Classes and Objects

```
java
CopyEdit
// Class with object creation

class Student {
    String name;
    int age;

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

public class ClassObjectExample {
    public static void main(String[] args) {
        Student s1 = new Student(); // Object
        s1.name = "Shantanu";
        s1.age = 20;
        s1.display();
    }
}
```

☑ 2. Constructors

```
java
CopyEdit
// Constructor Example

class Car {
    String model;

    // Constructor
    Car(String m) {
```

```
        model = m;
    }

    void show() {
        System.out.println("Model: " + model);
    }
}

public class ConstructorExample {
    public static void main(String[] args) {
        Car c1 = new Car("Tata Punch");
        c1.show();
    }
}
```

☑ 3. Method Overloading

```
java
CopyEdit
// Same method name, different parameters

class Calculator {
    int add(int a, int b) {
        return a + b;
    }

    float add(float a, float b) {
        return a + b;
    }
}

public class OverloadingExample {
    public static void main(String[] args) {
        Calculator calc = new Calculator();
        System.out.println("Int Add: " + calc.add(5, 3));
        System.out.println("Float Add: " + calc.add(2.5f, 1.5f));
    }
}
```

☑ 4. Static Keyword

```
java
CopyEdit
// Static variable and method

class Counter {
    static int count = 0;

    Counter() {
        count++;
        System.out.println("Object created. Count = " + count);
    }

    static void showCount() {
        System.out.println("Total objects: " + count);
    }
}
```

```
public class StaticExample {
    public static void main(String[] args) {
        new Counter();
        new Counter();
        Counter.showCount(); // static method
    }
}
```

☑ 5. `this` Keyword

```
java
CopyEdit
// Using 'this' to refer current object

class Person {
    String name;

    Person(String name) {
        this.name = name; // 'this' refers to instance variable
    }

    void show() {
        System.out.println("Name: " + name);
    }
}

public class ThisKeywordExample {
    public static void main(String[] args) {
        Person p = new Person("Shantanu");
        p.show();
    }
}
```

☑ 6. Garbage Collection

```
java
CopyEdit
// Java automatically removes unused objects

public class GarbageExample {
    public static void main(String[] args) {
        GarbageExample obj1 = new GarbageExample();
        obj1 = null; // eligible for garbage collection

        System.gc(); // Suggests JVM to run garbage collector
    }

    // finalize method runs before object is destroyed
    protected void finalize() {
        System.out.println("Object is garbage collected");
    }
}
```

☑ 7. Finalize Method

```
java
CopyEdit
// finalize() runs before object is destroyed

public class FinalizeExample {
    protected void finalize() {
        System.out.println("finalize() called");
    }

    public static void main(String[] args) {
        FinalizeExample f = new FinalizeExample();
        f = null;
        System.gc(); // request garbage collection
    }
}
```

☑ 8. Nested and Inner Classes

```
java
CopyEdit
// Inner class example

class Outer {
    int outerVar = 10;

    class Inner {
        void display() {
            System.out.println("Inner class accessing outerVar: " +
outerVar);
        }
    }
}

public class InnerClassExample {
    public static void main(String[] args) {
        Outer outer = new Outer();
        Outer.Inner inner = outer.new Inner();
        inner.display();
    }
}
```