OOP_PRACTICAL_CODES

1. Install Java and Print "Hello World"

Program :

```
public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello World!");
    }
}
```

2. Programs based on Basic Syntactic Constructs

Program :

A. Operators and Expressions

```
public class OperatorsExample {
    public static void main(String[] args) {
        int a = 10, b = 20;
        int sum = a + b;
        System.out.println("Sum: " + sum);
    }
}
```

B. Looping Statements

```
public class LoopExample {
    public static void main(String[] args) {
        for (int i = 1; i <= 5; i++) {
            System.out.println("Count: " + i);
        }
    }
}
```

C. Decision Making Statements

```
public class DecisionExample {
    public static void main(String[] args) {
        int num = 10;
        if (num > 0) {
            System.out.println("Positive number");
        } else {
            System.out.println("Negative number");
        } }}
```

3. Class, Constructors, Overloading

Program :

```
class Car {
    String model;
    int year;

    // Constructor
    Car(String model) {
        this.model = model;
    }

    // Constructor Overloading
    Car(String model, int year) {
        this.model = model;
        this.year = year;
    }

    void display() {
        System.out.println("Model: " + model + ", Year: " + year);
    }
}

public class ConstructorExample {
    public static void main(String[] args) {
        Car car1 = new Car("Toyota");
        Car car2 = new Car("Honda", 2022);
        car1.display();
```

```
    car2.display();
}}
```

4. Scanner Class Usage

Program :

```
import java.util.Scanner;

public class ScannerExample {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter your name: ");
        String name = sc.nextLine();
        System.out.println("Hello, " + name + "!");
    }
}
```

5. Inheritance and Access Control

Program :

```
class Animal {
    protected String name = "Animal";
    public void display() {
        System.out.println("I am an " + name);
    }
}

class Dog extends Animal {
    private String breed = "Labrador";
    public void showBreed() {
        System.out.println("Breed: " + breed);
    }
}

public class InheritanceExample {
    public static void main(String[] args) {
        Dog dog = new Dog();
        dog.display();
        dog.showBreed();
    }
}
```

6. Method Overriding

Program :

```
class Vehicle {
    public void run() {
        System.out.println("Vehicle is running");
    }
}

class Bike extends Vehicle {
    @Override
    public void run() {
        System.out.println("Bike is running safely");
    }
}

public class MethodOverridingExample {
    public static void main(String[] args) {
        Bike bike = new Bike();
        bike.run();
    }
}
```

7. Implementing Interfaces

Program :

```
interface Drawable {
    void draw();
}

class Circle implements Drawable {
    public void draw() {
        System.out.println("Drawing Circle");
    }
}

public class InterfaceExample {
    public static void main(String[] args) {
        Circle c = new Circle();
        c.draw();
    }
}
```

8. Using Object Class

Program :

```
class Student {
    int id;
    String name;

    Student(int id, String name) {
        this.id = id;
        this.name = name;
    }

    public String toString() {
        return id + " " + name;
    }
}

public class ObjectClassExample {
    public static void main(String[] args) {
        Student s1 = new Student(1, "Alice");
        System.out.println(s1);
    }
}
```

11. Packages and Sub-Packages

First create a package structure:

Create folder mypack

Inside mypack, file:

// mypack/PackageExample.java

Program :

```
package mypack;
public class PackageExample {
    public void display() {
        System.out.println("This is a package example");
    }
}
```

Main Program:

// MainPackage.java

```
import mypack.PackageExample;

public class MainPackage {
    public static void main(String[] args) {
        PackageExample obj = new PackageExample();
        obj.display();
    }
}
```

12. Applet Program

Program :

```
import java.applet.Applet;
import java.awt.Graphics;

/* <applet code="SimpleApplet" width="300" height="300"></applet> */
public class SimpleApplet extends Applet {
    public void paint(Graphics g) {
        g.drawString("Hello Applet!", 100, 150);
    }
}
```

13. Exception Handling

Program :

```
public class ExceptionHandlingExample {
    public static void main(String[] args) {
        try {
            int data = 50 / 0;
        } catch (ArithmeticException e) {
            System.out.println("Can't divide by zero");
        }
        System.out.println("Rest of the code...");
    }
}
```

14. Multithreading

Program :

```
class MyThread extends Thread {
    public void run() {
        for (int i = 1; i <= 5; i++) {
            System.out.println(i);
            try {
                Thread.sleep(500);
            } catch (InterruptedException e) {
                System.out.println(e);
            }
        }
    }
}

public class MultiThreadingExample {
    public static void main(String[] args) {
        MyThread t1 = new MyThread();
        t1.start();
    }
}
```