1. Hello World:

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

2. Basic Calculator:

```java
public class Calculator {
    public static void main(String[] args) {
        int num1 = 10, num2 = 5;
        int sum = num1 + num2;
        System.out.println("Sum: " + sum);
    }
}
```

3. Fibonacci Series:

```java
public class Fibonacci {
    public static void main(String[] args) {
        int n = 10;
        int a = 0, b = 1;

        System.out.print(a + " " + b);
        for (int i = 2; i < n; i++) {
            int next = a + b;
            System.out.print(" " + next);
            a = b;
            b = next;
        }
    }
}
```

4. Factorial Calculation:

```java
public class Factorial {
    public static void main(String[] args) {
```

        int n = 5;
        int factorial = 1;

        for (int i = 1; i <= n; i++) {
            factorial *= i;
        }

        System.out.println("Factorial of " + n + " is: " + factorial);
    }
}

5. Array Manipulation:
```java
public class ArrayExample {
    public static void main(String[] args) {
        int[] numbers = {5, 2, 8, 1, 9};
        int sum = 0;

        for (int num : numbers) {
            sum += num;
        }

        System.out.println("Sum of array elements: " + sum);
    }
}
```

6. Prime Number Check:
```java
public class PrimeCheck {
    public static void main(String[] args) {
        int num = 17;
        boolean isPrime = true;

        for (int i = 2; i <= Math.sqrt(num); i++) {
            if (num % i == 0) {
                isPrime = false;
                break;
            }
        }

        System.out.println(num + " is prime: " + isPrime);
    }
}
```

7. String Reversal:

```java
public class StringReverse {
    public static void main(String[] args) {
        String original = "Hello, World!";
        StringBuilder reversed = new StringBuilder();

        for (int i = original.length() - 1; i >= 0; i--) {
            reversed.append(original.charAt(i));
        }

        System.out.println("Reversed string: " + reversed.toString());
    }
}
```

8. Palindrome Check:

```java
public class PalindromeCheck {
    public static void main(String[] args) {
        String word = "radar";
        boolean isPalindrome = true;

        for (int i = 0; i < word.length() / 2; i++) {
            if (word.charAt(i) != word.charAt(word.length() - 1 - i)) {
                isPalindrome = false;
                break;
            }
        }

        System.out.println(word + " is palindrome: " + isPalindrome);
    }
}
```

9. Class and Object:

```java
class Student {
    String name;
    int age;
    void displayInfo() {
        System.out.println("Name: " + name);
        System.out.println("Age: " + age);
    }
}

public class StudentDemo {
    public static void main(String[] args) {
        Student student1 = new Student();
        student1.name = "John";
        student1.age = 20;
        student1.displayInfo();
    }
}
```

10. Inheritance:

```java
class Animal {
    void eat() {
        System.out.println("Animal is eating.");
    }
}

class Dog extends Animal {
    void bark() {
        System.out.println("Dog is barking.");
    }
}

public class InheritanceDemo {
    public static void main(String[] args) {
        Dog dog = new Dog();
        dog.eat();
        dog.bark();
    }
}
```

11. Exception Handling:

```java
public class ExceptionHandling {
    public static void main(String[] args) {
        try {
            int result = 10 / 0;
            System.out.println("Result: " + result);
        } catch (ArithmeticException e) {
            System.out.println("Error: " + e.getMessage());
        }
    }
}
```

12. File Handling:

```java
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;

public class FileHandling {
    public static void main(String[] args) {
        try {
            File file = new File("example.txt");
            FileWriter writer = new FileWriter(file);
            writer.write("Hello, File Handling!");
            writer.close();
        } catch (IOException e) {
            System.out.println("Error: " + e.getMessage());
        }
    }
}
```

13. Multithreading:

```java
class MyThread extends Thread {
    public void run() {
        for (int i = 1; i <= 5; i++) {
            System.out.println(Thread.currentThread().getName() + " Value " + i);
        }
    }
}

public class MultithreadingDemo {
    public static void main(String[] args) {
        MyThread t1 = new MyThread();
        MyThread t2 = new MyThread();
        t1.start();
```

        t2.start();
    }
}

14. ArrayList Usage:
```java
import java.util.ArrayList;

public class ArrayListDemo {
    public static void main(String[] args) {
        ArrayList<String> fruits = new ArrayList<>();
        fruits.add("Apple");
        fruits.add("Banana");
        fruits.add("Orange");

        System.out.println("Fruits: " + fruits);
    }
}
```

15. Interface Implementation:
```java
interface Shape {
    double getArea();
}

class Circle implements Shape {
    double radius;

    Circle(double radius) {
        this.radius = radius;
    }

    public double getArea() {
        return Math.PI * radius * radius;
    }
}

public class InterfaceDemo {
    public static void main(String[] args) {
        Circle circle = new Circle(5);
        System.out.println("Area of circle: " + circle.getArea());
    }
```

~~~
}
```
~~~

Feel free to explore and modify these sample programs to learn more about Java programming concepts!