Java Basic Coding Challenges

1.Primitive Data types

1. Declare and initialize primitive types and print their default values

```
public class PrimitiveDefaults {
byte byteVar;
short shortVar;
int intVar;
long longVar;
float floatVar;
double doubleVar;
char charVar;
boolean booleanVar;
public void printDefaults() {
        System.out.println("Default byte: " + byteVar);
        System.out.println("Default short: " + shortVar);
        System.out.println("Default int: " + intVar);
        System.out.println("Default long: " + longVar);
        System.out.println("Default float: " + floatVar);
        System.out.println("Default double: " + doubleVar);
        System.out.println("Default char: [" + charVar + "]");
        System.out.println("Default boolean: " + booleanVar);
}
public static void main(String[] args) {
        PrimitiveDefaults obj = new PrimitiveDefaults();
        obj.printDefaults();
}
}
```

2. Detect overflow when adding two byte variables

```
public class ByteOverflowDetection {
        public static void main(String[] args) {
                byte a = 120;
                byte b = 10;
                int result = a + b;
                if (result > Byte.MAX_VALUE || result < Byte.MIN_VALUE) {
                System.out.println("Overflow detected! Result = " + result);
                } else {
                byte sum = (byte) result;
                System.out.println("No overflow. Sum = " + sum);
                }
        }
}
```

3. Type casting double to int and float to byte

```
public class TypeCastingExample {
public static void main(String[] args) {
double doubleValue = 123.456;
float floatValue = 130.75f;
int intValue = (int) doubleValue;
byte byteValue = (byte) floatValue;
System.out.println("Double to int: " + intValue);
System.out.println("Float to byte: " + byteValue);
}
}
```

4. Bitwise operations between int and byte

```
public class BitwiseOperations {
public static void main(String[] args) {
byte b = 0b0101;
int i = 0b00110000;
System.out.println("AND: " + (b & i));
System.out.println("OR: " + (b | i));
System.out.println("XOR: " + (b ^ i));
System.out.println("NOT b: " + (~b));
}
}
```

5. Accept input for all primitive types and display formatted

```
import java.util.Scanner;
public class PrimitiveInput {
public static void main(String[] args) {
Scanner sc = new Scanner(System.in);
byte b = sc.nextByte();
short s = sc.nextShort();
int i = sc.nextInt();
long l = sc.nextLong();
float f = sc.nextFloat();
double d = sc.nextDouble();
char c = sc.next().charAt(0);
boolean bool = sc.nextBoolean();
System.out.printf("Byte: %d, Short: %d, Int: %d, Long: %d\n", b, s, i, l);
System.out.printf("Float: %.2f, Double: %.3f\n", f, d);
System.out.println("Char: " + c + ", Boolean: " + bool);
}
}
```

2)Variables

1. Swap two numbers using a temporary variable

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

2. Swap two numbers without using a temporary variable

```
public class SwapWithoutTemp {
        public static void main(String[] args) {
        int a = 5, b = 7;
        a = a + b;
        b = a - b;
        a = a - b;
        System.out.println("a = " + a + ", b = " + b);
        }
}
```

3. Demonstrate variable shadowing within a class and method

```
public class VariableShadowing {
int number = 100;
        public void display() {
        int number = 50;
        System.out.println("Local: " + number);
        System.out.println("Instance: " + this.number);
        }
        public static void main(String[] args) {
                new VariableShadowing().display();
        }
}
```

4. Declare a constant and use it in calculations

```
public class ConstantUsage {
        public static final double PI = 3.14159;
        public static void main(String[] args) {
                double radius = 5.0;
                double area = PI * radius * radius;
                System.out.println("Area = " + area);
        }
}
```

5. Create a class with instance, static, and local variables and demonstrate scope

```
public class ScopeDemo {
    int instanceVar = 10;
    static int staticVar = 20;
    public void show() {
    int localVar = 30;
    System.out.println("Instance: " + instanceVar);
    System.out.println("Static: " + staticVar);
    System.out.println("Local: " + localVar);
}
public static void main(String[] args) {
    ScopeDemo obj1 = new ScopeDemo();
    obj1.instanceVar = 100;
    ScopeDemo obj2 = new ScopeDemo();
    obj2.instanceVar = 200;
    ScopeDemo.staticVar = 300;
    obj1.show();
    obj2.show();
    }
}
```

3.Operators

1. Demonstrate all arithmetic operators using two integers

```
public class ArithmeticOperators {
    public static void main(String[] args) {
        int a = 15, b = 4;
        System.out.println("a + b = " + (a + b));
        System.out.println("a - b = " + (a - b));
        System.out.println("a * b = " + (a * b));
        System.out.println("a / b = " + (a / b));
        System.out.println("a % b = " + (a % b));
    }
}
```

2. Use relational operators to compare ages

```
import java.util.Scanner;
public class AgeComparison {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter age of Person A: ");
        int ageA = sc.nextInt();
        System.out.print("Enter age of Person B: ");
        int ageB = sc.nextInt();
        System.out.println("A == B: " + (ageA == ageB));
        System.out.println("A != B: " + (ageA != ageB));
        System.out.println("A > B: " + (ageA > ageB));
        System.out.println("A < B: " + (ageA < ageB));
        System.out.println("A >= B: " + (ageA >= ageB));
        System.out.println("A <= B: " + (ageA <= ageB));
    }
}
```

3. Implement a basic calculator using switch and operators

```
import java.util.Scanner;
public class BasicCalculator {
    public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    System.out.print("Enter first number: ");
    double num1 = sc.nextDouble();
    System.out.print("Enter operator (+, -, *, /, %): ");
    char op = sc.next().charAt(0);
    System.out.print("Enter second number: ");
    double num2 = sc.nextDouble();
    switch (op) {
        case '+': System.out.println("Result = " + (num1 + num2)); break;
        case '-': System.out.println("Result = " + (num1 - num2)); break;
        case '*': System.out.println("Result = " + (num1 * num2)); break;
        case '/':
        if (num2 != 0) System.out.println("Result = " + (num1 / num2));
        else System.out.println("Cannot divide by zero");
        break;
        case '%':
        if (num2 != 0) System.out.println("Result = " + (num1 % num2));
        else System.out.println("Cannot mod by zero");
        break;
        default: System.out.println("Invalid operator");
    }
    }
}
```

4. Use bitwise AND, OR, XOR on two binary values

```
public class BitwiseBinary {
    public static void main(String[] args) {
        int a = 0b1100;
        int b = 0b1010;
        System.out.println("a & b = " + Integer.toBinaryString(a & b));
        System.out.println("a | b = " + Integer.toBinaryString(a | b));
        System.out.println("a ^ b = " + Integer.toBinaryString(a ^ b));
    }
}
```

5. Demonstrate logical operators with Boolean expressions

```
public class LogicalOperatorsDemo {
    public static void main(String[] args) {
        boolean a = true, b = false;
        System.out.println("a && b = " + (a && b));
        System.out.println("a || b = " + (a || b));
        System.out.println("!a = " + (!a));
        int age = 20;
        boolean hasID = true;
        System.out.println("Can enter club: " + (age >= 18 && hasID));
        }
}
```

4)String Concatenation

1. Concatenate first name and last name

```
import java.util.Scanner;
public class NameConcatenation {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter first name: ");
        String firstName = scanner.nextLine();
        System.out.print("Enter last name: ");
        String lastName = scanner.nextLine();
        String fullName = firstName + " " + lastName;
        System.out.println("Full Name: " + fullName);
        scanner.close();
    }
}
```

2. Combine name, age, and address using string concatenation

```
import java.util.Scanner;
public class PersonalInfoConcatenation {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter your name: ");
        String name = scanner.nextLine();
        System.out.print("Enter your age: ");
        int age = scanner.nextInt();
        scanner.nextLine(); // consume leftover newline
        System.out.print("Enter your address: ");
        String address = scanner.nextLine();
        String info = "Name: " + name + ", Age: " + age + ", Address: " + address;
        System.out.println(info);
        scanner.close();
    }
}
```

3. Use concatenation inside a loop to build a pattern

```
import java.util.Scanner;
public class PatternConcatenation {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter number of rows: ");
        int rows = scanner.nextInt();
        String pattern = "";
        for (int i = 1; i <= rows; i++) {
        pattern += "*";
        System.out.println(pattern);
        }
        scanner.close();
  }
}
```

4. Demonstrate precedence of concatenation and addition

```
public class ConcatPrecedence {
    public static void main(String[] args) {
        String name = "Alice";
        System.out.println(1 + 2 + name); // 3Alice -> left-to-right, 1+2 first
        System.out.println(name + 1 + 2); // Alice12 -> after String, rest are
        concatenated
        System.out.println(name + (1 + 2)); // Alice3 -> parentheses force
        addition
        System.out.println(1 + (2 + name)); // 12Alice -> 2+name -> "2Alice", then
        1 + "2Alice"
        System.out.println("" + 1 + 2 + name); // 12Alice -> starting with "" forces
        all to String
    }
}
```

5. Accept input strings and concatenate with formatting

```
import java.util.Scanner;
public class FormattedConcatenation {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter city: ");
        String city = sc.nextLine();
        System.out.print("Enter state: ");
        String state = sc.nextLine();
        System.out.print("Enter country: ");
        String country = sc.nextLine();
        // Using concatenation
        String address1 = city + ", " + state + ", " + country;
        // Using String.format
        String address2 = String.format("%s, %s, %s", city, state, country);
        System.out.println("Concatenated: " + address1);
        System.out.println("Formatted : " + address2);
        sc.close();
    }
}
```

5)StringBuilder

1. Challenge: Reverse a string using StringBuilder

```
public class ReverseString {
        public static void main(String[] args) {
                String input = "OpenAI";
                StringBuilder sb = new StringBuilder(input);
                System.out.println("Reversed: " + sb.reverse());
        }
}
```

2. Challenge: Append multiple strings using StringBuilder and print

```
public class AppendStrings {
        public static void main(String[] args) {
                StringBuilder sb = new StringBuilder();
                sb.append("Hello, ");
                sb.append("this is ");
                sb.append("a concatenated string.");
                System.out.println(sb.toString());
        }
}
```

3. Challenge: Replace characters in a string using StringBuilder

```
public class ReplaceCharacters {
        public static void main(String[] args) {
                StringBuilder sb = new StringBuilder("hello world");
                sb.setCharAt(6, 'W'); // Replacing 'w' with 'W'
                System.out.println("Modified: " + sb.toString());
        }
}
```

4. Challenge: Insert a word into a string at a specific position

```
public class InsertWord {
        public static void main(String[] args) {
                StringBuilder sb = new StringBuilder("I Java");
                sb.insert(2, "love "); // Inserting at index 2
                System.out.println(sb.toString()); // Output: I love Java
```

```
    }
}
```

5. Challenge: Delete part of a string using StringBuilder

```
public class DeleteSubstring {
public static void main(String[] args) {
StringBuilder sb = new StringBuilder("RemoveThisPart");
sb.delete(0, 6); // Deletes "Remove"
System.out.println("After deletion: " + sb.toString()); // Output: ThisPart
}
}
```

6)String API’s

1. Challenge: Use charAt(), length(), and substring() methods

```
public class StringMethods {
    public static void main(String[] args) {
        String text = "HelloWorld";
        System.out.println("Char at index 1: " + text.charAt(1));
        System.out.println("Length: " + text.length());
        System.out.println("Substring (0, 5): " + text.substring(0, 5));
    }
}
```

2. Challenge: Count the number of vowels in a string

```
public class CountVowels {
    public static void main(String[] args) {
        String str = "OpenAI ChatGPT";
        int count = 0;
        for (char c : str.toLowerCase().toCharArray()) {
        if ("aeiou".indexOf(c) != -1) {
        count++;
    }
}
System.out.println("Number of vowels: " + count);
}
}
```

3. Challenge: Check if a string is a palindrome

```
public class PalindromeCheck {
    public static void main(String[] args) {
        String original = "madam";
        String reversed = new StringBuilder(original).reverse().toString();
        if (original.equals(reversed)) {
        System.out.println(original + " is a palindrome.");
        } else {
        System.out.println(original + " is not a palindrome.");
        }
    }
}
```

4. Challenge: Convert a string to upper case and lower case

```
public class CaseConversion {
    public static void main(String[] args) {
        String text = "JavaProgramming";
        System.out.println("Upper Case: " + text.toUpperCase());
        System.out.println("Lower Case: " + text.toLowerCase());
    }
}
```

5. Challenge: Remove spaces and special characters from a string

```
public class CleanString {
    public static void main(String[] args) {
    String str = "Hello @World! 123";
    String cleaned = str.replaceAll("[^a-zA-Z0-9]", "");
    System.out.println("Cleaned String: " + cleaned);
    }
}
```

7)Date, Time, and Numeric Objects

1. Challenge: Get current date and time using LocalDateTime

```
import java.time.LocalDateTime;
public class CurrentDateTime {
        public static void main(String[] args) {
        LocalDateTime now = LocalDateTime.now();
        System.out.println("Current Date and Time: " + now);
        }
}
```

2. Challenge: Calculate age given a birth date

```
import java.time.LocalDate;
import java.time.Period;
public class CalculateAge {
        public static void main(String[] args) {
        LocalDate birthDate = LocalDate.of(2000, 1, 1);
        LocalDate today = LocalDate.now();
        Period age = Period.between(birthDate, today);
        System.out.println("Age: " + age.getYears() + " years");
        }
}
```

3. Challenge: Format date in dd-MM-yyyy format

```
import java.time.LocalDate;
import java.time.format.DateTimeFormatter;
public class FormatDate {
        public static void main(String[] args) {
        LocalDate date = LocalDate.now();
        DateTimeFormatter formatter = DateTimeFormatter.ofPattern("dd-MM-yyyy");
        System.out.println("Formatted Date: " + date.format(formatter));
        }
}
```

4. Challenge: Add 5 days to current date and print

```
import java.time.LocalDate;
public class AddDays {
	public static void main(String[] args) {
	LocalDate today = LocalDate.now();
	LocalDate futureDate = today.plusDays(5);
	System.out.println("Date after 5 days: " + futureDate);
	}
}
```

5. Challenge: Round a decimal to 2 places using BigDecimal

```
import java.math.BigDecimal;
import java.math.RoundingMode;
public class RoundDecimal {
	public static void main(String[] args) {
	BigDecimal number = new BigDecimal("123.456789");
	BigDecimal rounded = number.setScale(2, RoundingMode.HALF_UP);
	System.out.println("Rounded: " + rounded);
	}
}
```

8)Flow Control Statements

1. Use if-else to determine if a number is positive, negative, or zero

```
import java.util.Scanner;
public class NumberSign {
	public static void main(String[] args) {
	Scanner sc = new Scanner(System.in);
	System.out.print("Enter a number: ");
	int number = sc.nextInt();
	if (number > 0)
	System.out.println("Positive");
	else if (number < 0)
	System.out.println("Negative");
	else
```

```
    System.out.println("Zero");
    }
}
```

2. Implement nested if to find the largest among 3 numbers

```
import java.util.Scanner;
public class LargestOfThree {
    public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    System.out.print("Enter three numbers: ");
    int a = sc.nextInt();
    int b = sc.nextInt();
    int c = sc.nextInt();
    if (a > b) {
    if (a > c)
        System.out.println("Largest is: " + a);
    else
        System.out.println("Largest is: " + c);
    } else {
    if (b > c)
        System.out.println("Largest is: " + b);
    else
        System.out.println("Largest is: " + c);
    }
}
}
```

3. Validate login with username and password

```
import java.util.Scanner;
public class LoginValidation {
    public static void main(String[] args) {
    final String USERNAME = "admin";
    final String PASSWORD = "1234";
    Scanner sc = new Scanner(System.in);
    System.out.print("Username: ");
    String user = sc.nextLine();
    System.out.print("Password: ");
    String pass = sc.nextLine();
    if (user.equals(USERNAME) && pass.equals(PASSWORD)) {
    System.out.println("Login successful");
    } else {
    System.out.println("Invalid username or password");
    }
    }
}
```

4. Categorize age groups using if-else ladder

```
import java.util.Scanner;
public class AgeGroup {
    public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    System.out.print("Enter age: ");
    int age = sc.nextInt();
    if (age < 13)
        System.out.println("Child");
    else if (age < 20)
        System.out.println("Teenager");
    else if (age < 60)
        System.out.println("Adult");
    else
        System.out.println("Senior");
    }
}
```

5. Determine student grade using percentage

```
import java.util.Scanner;
```

```
public class StudentGrade {
        public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter percentage: ");
        double percent = sc.nextDouble();
        if (percent >= 90)
                System.out.println("Grade: A");
        else if (percent >= 80)
                System.out.println("Grade: B");
        else if (percent >= 70)
                System.out.println("Grade: C");
        else if (percent >= 60)
                System.out.println("Grade: D");
        else
                System.out.println("Grade: F");
        }
}
```

9)Conditions

1. Check if a number is even and divisible by 5

```
import java.util.Scanner;
public class EvenAndDivisible {
        public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int num = sc.nextInt();
        if (num % 2 == 0 && num % 5 == 0)
        System.out.println("The number is even and divisible by 5.");
        else
        System.out.println("The number does not meet both conditions.");
        }
}
```

2. Validate a triangle (sum of angles = 180)

```
import java.util.Scanner;
    public class TriangleValidation {
    public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    System.out.print("Enter three angles: ");
    int a = sc.nextInt();
    int b = sc.nextInt();
    int c = sc.nextInt();
    if (a + b + c == 180)
    System.out.println("Valid triangle");
    else
    System.out.println("Invalid triangle");
    }
}
```

3. Check if year is a leap year

```
import java.util.Scanner;
public class LeapYearCheck {
    public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    System.out.print("Enter a year: ");
    int year = sc.nextInt();
    if ((year % 4 == 0 && year % 100 != 0) || (year % 400 == 0))
    System.out.println("Leap year");
    else
    System.out.println("Not a leap year");
    }
}
```

4. Check character type (vowel/consonant/digit/special)

```
import java.util.Scanner;
public class CharacterType {
public static void main(String[] args) {
Scanner sc = new Scanner(System.in);
System.out.print("Enter a character: ");
char ch = sc.next().charAt(0);
if (Character.isDigit(ch))
    System.out.println("Digit");
else if (Character.isLetter(ch)) {
    ch = Character.toLowerCase(ch);
if ("aeiou".indexOf(ch) != -1)
    System.out.println("Vowel");
else
    System.out.println("Consonant");
} else
    System.out.println("Special character");
}
}
```

5. Check eligibility for vote, driving, and job using conditions

```
import java.util.Scanner;
public class EligibilityCheck {
    public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    System.out.print("Enter age: ");
    int age = sc.nextInt();
    if (age >= 18)
        System.out.println("Eligible to vote");
    else
        System.out.println("Not eligible to vote");
    if (age >= 16)
        System.out.println("Eligible to drive");
    else
        System.out.println("Not eligible to drive");
        if (age >= 21)
                System.out.println("Eligible for job");
        else
                System.out.println("Not eligible for job");
        }
}
```

10)Switch

1. Create a calculator using switch

```
import java.util.Scanner;
public class SwitchCalculator {
        public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter two numbers: ");
        double a = sc.nextDouble();
        double b = sc.nextDouble();
        System.out.print("Enter operator (+, -, *, /): ");
        char op = sc.next().charAt(0);
        switch (op) {
        case '+':
                System.out.println("Result: " + (a + b));
                break;
        case '-':
                System.out.println("Result: " + (a - b));
                break;
        case '*':
                System.out.println("Result: " + (a * b));
                break;
        case '/':
                if (b != 0)
                System.out.println("Result: " + (a / b));
                else
        System.out.println("Cannot divide by zero");
        break;
    default:
        System.out.println("Invalid operator");
        break;
    }
    }
}
```

2. Map number to month name using switch

```
import java.util.Scanner;
public class MonthMapper {
public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    System.out.print("Enter month number (1-12): ");
    int month = sc.nextInt();
    switch (month) {
        case 1: System.out.println("January"); break;
        case 2: System.out.println("February"); break;
        case 3: System.out.println("March"); break;
        case 4: System.out.println("April"); break;
        case 5: System.out.println("May"); break;
        case 6: System.out.println("June"); break;
        case 7: System.out.println("July"); break;
        case 8: System.out.println("August"); break;
        case 9: System.out.println("September"); break;
        case 10: System.out.println("October"); break;
        case 11: System.out.println("November"); break;
        case 12: System.out.println("December"); break;
        default: System.out.println("Invalid month number");
    }
    }
}
```

3. Implement a simple menu using switch

```
import java.util.Scanner;
public class MenuExample {
public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Menu:");
        System.out.println("1. Say Hello");
        System.out.println("2. Say Goodbye");
        System.out.println("3. Exit");
        System.out.print("Enter choice: ");
        int choice = sc.nextInt();
        switch (choice) {
        case 1:
        System.out.println("Hello!");
        break;
        case 2:
        System.out.println("Goodbye!");
        break;
        case 3:
        System.out.println("Exiting...");
        break;
        default:
        System.out.println("Invalid choice");
        }
        }
}
```

4. Use enhanced switch (Java 14+) for better syntax

```
import java.util.Scanner;
public class EnhancedSwitch {
public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    System.out.print("Enter a day number (1-7): ");
    int day = sc.nextInt();
    String dayName = switch (day) {
    case 1 -> "Monday";
    case 2 -> "Tuesday";
    case 3 -> "Wednesday";
    case 4 -> "Thursday";
    case 5 -> "Friday";
    case 6 -> "Saturday";
    case 7 -> "Sunday";
    default -> "Invalid day";
    };
    System.out.println("Day: " + dayName);
    }
}
```

5. Implement day of the week based on integer input

```
import java.util.Scanner;
public class DayOfWeek {
public static void main(String[] args) {
Scanner sc = new Scanner(System.in);
System.out.print("Enter day number (1-7): ");
int day = sc.nextInt();
switch (day) {
case 1: System.out.println("Sunday"); break;
case 2: System.out.println("Monday"); break;
case 3: System.out.println("Tuesday"); break;
case 4: System.out.println("Wednesday"); break;
case 5: System.out.println("Thursday"); break;
case 6: System.out.println("Friday"); break;
case 7: System.out.println("Saturday"); break;
default: System.out.println("Invalid input");
}
}
}
```

11)Loops and Branching

1. Print multiplication table for a number

```
import java.util.Scanner;
public class MultiplicationTable {
	public static void main(String[] args) {
		Scanner sc = new Scanner(System.in);
		System.out.print("Enter a number: ");
		int num = sc.nextInt();
		for (int i = 1; i <= 10; i++) {
		System.out.println(num + " x " + i + " = " + (num * i));
		}
	}
}
```

2. Use break and continue in loops

```
public class BreakContinueExample {
public static void main(String[] args) {
for (int i = 1; i <= 10; i++) {
	if (i == 5)
		continue;
	if (i == 8)
		break;
	System.out.println(i);
}
}
}
```

3. Find factorial of a number

```
import java.util.Scanner;
```

```
public class Factorial {
        public static void main(String[] args) {
                Scanner sc = new Scanner(System.in);
                System.out.print("Enter a number: ");
                int num = sc.nextInt();
                long fact = 1;
                for (int i = 1; i <= num; i++) {
                fact *= i;
                }
                System.out.println("Factorial: " + fact);
        }
}
```

4. Print Fibonacci series

```
import java.util.Scanner;
public class FibonacciSeries {
public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter count: ");
        int count = sc.nextInt();
        int a = 0, b = 1;
        System.out.print("Fibonacci: " + a + " " + b + " ");
        for (int i = 2; i < count; i++) {
        int c = a + b;
        System.out.print(c + " ");
        a = b;
        b = c;
        }
        }
}
```

5. Find sum of even numbers from 1 to 100

```
public class SumEvenNumbers {
```

```
public static void main(String[] args) {
int sum = 0;
for (int i = 2; i <= 100; i += 2) {
sum += i;
}
System.out.println("Sum of even numbers from 1 to 100: " + sum);
}
}
```

12)Arrays

1. Find the largest and smallest element in an array

```
import java.util.Scanner;
public class MinMaxArray {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter array size: ");
        int n = sc.nextInt();
        int[] arr = new int[n];
        System.out.println("Enter elements:");
        for (int i = 0; i < n; i++)
        arr[i] = sc.nextInt();
        int min = arr[0], max = arr[0];
        for (int i = 1; i < n; i++) {
        if (arr[i] < min) min = arr[i];
        if (arr[i] > max) max = arr[i];
    }
    System.out.println("Min: " + min + ", Max: " + max);
    }
}
```

2. Sort an array in ascending order

```
import java.util.Arrays;
import java.util.Scanner;
```

```
public class SortArray {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter array size: ");
        int n = sc.nextInt();
        int[] arr = new int[n];
        System.out.println("Enter elements:");
        for (int i = 0; i < n; i++)
            arr[i] = sc.nextInt();
        Arrays.sort(arr);
        System.out.println("Sorted array:");
        for (int num : arr)
        System.out.print(num + " ");
    }
}
```

3. Calculate average of numbers in an array

```
import java.util.Scanner;
public class ArrayAverage {
public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    System.out.print("Enter array size: ");
    int n = sc.nextInt();
    int[] arr = new int[n];
    System.out.println("Enter elements:");
    for (int i = 0; i < n; i++)
        arr[i] = sc.nextInt();
    int sum = 0;
    for (int num : arr)
    sum += num;
    double avg = (double) sum / n;
    System.out.println("Average: " + avg);
    }
}
```

4. Count occurrence of an element

```
import java.util.Scanner;
public class CountOccurrence {
    public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    System.out.print("Enter array size: ");
    int n = sc.nextInt();
    int[] arr = new int[n];
    System.out.println("Enter elements:");
    for (int i = 0; i < n; i++)
        arr[i] = sc.nextInt();
    System.out.print("Enter number to count: ");
    int x = sc.nextInt();
    int count = 0;
    for (int num : arr)
        if (num == x) count++;
        System.out.println(x + " occurred " + count + " times.");
    }
}
```

5. Reverse elements of an array

```
import java.util.Scanner;
public class ReverseArray {
    public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    System.out.print("Enter array size: ");
    int n = sc.nextInt();
    int[] arr = new int[n];
    System.out.println("Enter elements:");
    for (int i = 0; i < n; i++)
    arr[i] = sc.nextInt();
    System.out.println("Reversed array:");
    for (int i = n - 1; i >= 0; i--)
    System.out.print(arr[i] + " ");
    }
}
```

13)Enum

1. Define an enum for days of the week

```
public class DaysOfWeekEnum {
enum Day {
SUNDAY, MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, SATURDAY
}
public static void main(String[] args) {
for (Day d : Day.values()) {
System.out.println(d);
}
}
}
```

2. Use enum in switch case

```
public class EnumSwitch {
enum Day {
MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, SATURDAY, SUNDAY
}
public static void main(String[] args) {
Day today = Day.WEDNESDAY;
switch (today) {
case MONDAY -> System.out.println("Start of the week!");
case FRIDAY -> System.out.println("Almost weekend!");
case SUNDAY -> System.out.println("Rest day!");
default -> System.out.println("Midweek day");
}
}
}
```

3. Iterate over enum values

```
public class EnumIteration {
    enum Season {
    SPRING, SUMMER, FALL, WINTER
    }
    public static void main(String[] args) {
    for (Season s : Season.values()) {
    System.out.println(s);
    }
    }
}
```

4. Assign properties to enum constants

```
public class EnumWithProperties {
    enum Planet {
    MERCURY(3.303e+23, 2.4397e6),
    VENUS(4.869e+24, 6.0518e6),
    EARTH(5.976e+24, 6.37814e6);
    private final double mass;
    private final double radius;
    Planet(double mass, double radius) {
    this.mass = mass;
    this.radius = radius;
    }
    public double surfaceGravity() {
    final double G = 6.67300E-11;
    return G * mass / (radius * radius);
}
public double getMass() { return mass; }
public double getRadius() { return radius; }
}
public static void main(String[] args) {
for (Planet p : Planet.values()) {
System.out.printf("%s: mass = %.2e, radius = %.2f%n", p, p.getMass(),
p.getRadius());
}
}
}
```

5. Create an enum to represent traffic light states

```
public class TrafficLightEnum {
        enum TrafficLight {
        RED, YELLOW, GREEN
        }
        public static void main(String[] args) {
        TrafficLight signal = TrafficLight.RED;
        switch (signal) {
        case RED -> System.out.println("Stop");
        case YELLOW -> System.out.println("Get Ready");
        case GREEN -> System.out.println("Go");
        }
    }
}
```