Java Basics Coding Challenges

1. Primitive Data Types

1.Challenge: Declare and initialize variables of all primitive types and print their default values.

```
public class PrimitiveDefaults {
    byte b = 0;
    short s = 0;
    int i = 0;
    long l = 0L;
    float f = 0.0f;
    double d = 0.0;
    char c = '\u0000';
    boolean bool = false;

    public static void main(String[] args) {
        PrimitiveDefaults p = new PrimitiveDefaults();
        System.out.println("byte: " + p.b);
        System.out.println("short: " + p.s);
        System.out.println("int: " + p.i);
        System.out.println("long: " + p.l);
        System.out.println("float: " + p.f);
        System.out.println("double: " + p.d);
        System.out.println("char: '" + p.c + "'");
        System.out.println("boolean: " + p.bool);
    }
}
```

2.Challenge: Write a program to detect overflow when adding two byte variables.

```
public class ByteOverflow {
    public static void main(String[] args) {
        byte a = 100;
        byte b = 15;
        byte result = (byte) (a + b);
        System.out.println("Overflow result: " + result);
    }
}
```

3.Challenge: Use type casting to convert double to int and float to byte.

```
public class TypeCasting {
    public static void main(String[] args) {
        double d = 123.456;
        int i = (int) d;
        float f = 123.45f;
        byte b = (byte) f;

        System.out.println("double to int: " + i);
        System.out.println("float to byte: " + b);
    }
}
```

4.Challenge: Perform bitwise operations between int and byte.

```
public class BitwiseOps {
    public static void main(String[] args) {
        int a = 5;
        byte b = 3;
        System.out.println("AND: " + (a & b));
        System.out.println("OR: " + (a | b));
        System.out.println("XOR: " + (a ^ b));
    }
}
```

```
              OUTPUT:
              AND: 1
              OR: 7
              XOR: 6
```

5.Challenge: Accept input for all primitive types and display them formatted.

```
import java.util.Scanner;

public class AllPrimitiveInput {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("Enter byte: ");
        byte b = sc.nextByte();

        System.out.print("Enter short: ");
        short s = sc.nextShort();

        System.out.print("Enter int: ");
        int i = sc.nextInt();

        System.out.print("Enter long: ");
        long l = sc.nextLong();

        System.out.print("Enter float: ");
        float f = sc.nextFloat();

        System.out.print("Enter double: ");
        double d = sc.nextDouble();

        System.out.print("Enter char: ");
        char c = sc.next().charAt(0);

        System.out.print("Enter boolean: ");
        boolean bool = sc.nextBoolean();
        System.out.printf("byte=%d, short=%d, int=%d, long=%d, float=%.2f, double=%.2f, char=%c, 
boolean=%b\n",
                b, s, i, l, f, d, c, bool);
    }
}
```

2. Variables

1.Challenge: Swap two numbers using a temporary variable.

```
public class SwapWithTemp {
    public static void main(String[] args) {
        int a = 4, b = 3;
        System.out.println("Before Swap: a=" + a + ", b=" + b);

        int temp = a;
        a = b;
        b = temp;

        System.out.println("After Swap: a=" + a + ", b=" + b);
    }
}
```

2.Challenge: Swap two numbers without a temporary variable.

```
public class SwapWithoutTemp {
    public static void main(String[] args) {
        int a = 5, b = 10;
        System.out.println("Before swap: a=" + a + ", b=" + b);

        a = a + b;
        b = a - b;
        a = a - b;

        System.out.println("After swap: a=" + a + ", b=" + b);
    }
}
```

3.Challenge: Demonstrate variable shadowing within a class and method.

```
public class VariableShadowing {
    int x = 100;

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

4.Challenge: Declare a constant and use it in calculations.

```
public class ConstantDemo {
    public static void main(String[] args) {
        final double PI = 3.14159;
        int radius = 7;
        double area = PI * radius * radius;

        System.out.println("Area of circle: " + area);
    }
}
```

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

```
public class VariableScopeDemo {
    int instanceVar = 10;
    static int staticVar = 20;

    public void show() {
        int localVar = 30;
        System.out.println("Instance variable: " + instanceVar);
        System.out.println("Static variable: " + staticVar);
        System.out.println("Local variable: " + localVar);
    }

    public static void main(String[] args) {
        VariableScopeDemo obj = new VariableScopeDemo();
        obj.show();
    }
}
```

3. Operators

1.Challenge: Demonstrate all arithmetic operators using two integers.

```
public class ArithmeticOperators {
    public static void main(String[] args) {
        int a = 10, b = 3;

        System.out.println("Addition: " + (a + b));
        System.out.println("Subtraction: " + (a - b));
        System.out.println("Multiplication: " + (a * b));
        System.out.println("Division: " + (a / b));
        System.out.println("Modulus: " + (a % b));
        }
}
```

2.Challenge: Use relational operators to compare ages.

```
public class RelationalOperators {
    public static void main(String[] args) {
        int age1 = 20, age2 = 25;

        System.out.println("age1 > age2: " + (age1 > age2));
        System.out.println("age1 < age2: " + (age1 < age2));
        System.out.println("age1 == age2: " + (age1 == age2));
        System.out.println("age1 != age2: " + (age1 != age2));
    }
}
```

3.Challenge: 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 a = sc.nextDouble();
        System.out.print("Enter second number: ");
        double b = sc.nextDouble();
        System.out.print("Enter operator (+, -, *, /): ");
        char op = sc.next().charAt(0);

        double result = 0;

        switch (op) {
            case '+': result = a + b; break;
            case '-': result = a - b; break;
            case '*': result = a * b; break;
            case '/':
                if (b != 0) result = a / b;
                else {
                    System.out.println("Error: Division by zero");
                    return;
                }
                break;
            default:
                System.out.println("Invalid operator");
                return;
        }

        System.out.println("Result: " + result);
    }
}
```

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

```
public class BitwiseOperators {
   public static void main(String[] args) {
      int a = 5;
      int b = 3;

      System.out.println("AND: " + (a & b));
      System.out.println("OR: " + (a | b));
      System.out.println("XOR: " + (a ^ b));
   }
}
```

5.Challenge: Demonstrate logical operators with Boolean expressions.

```
public class LogicalOperators {
             public static void main(String[] args) {
                boolean a = true, b = false;

                System.out.println("AND (a && b): " + (a && b)); // false
                System.out.println("OR  (a || b): " + (a || b)); // true
                System.out.println("NOT (!a): " + (!a));      // false
             }
          }
```

4. String Concatenation

1.Challenge: Concatenate first name and last name.

```
public class FullName {
   public static void main(String[] args) {
      String firstName = "G";
      String lastName = "Kavita";
      String fullName = firstName + " " + lastName;

      System.out.println("Full Name: " + fullName);
   }
}
```

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

```
public class UserInfo {
   public static void main(String[] args) {
      String name = "A vimala";
      int age = 23;
      String address = "Bangalore";

      String info = "Name: " + name + ", Age: " + age + ", Address: " + address;
      System.out.println(info);
   }
}
```

3.Challenge: Use concatenation inside a loop to build a pattern.

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

4.Challenge: Demonstrate precedence of concatenation and addition.

```
public class PrecedenceDemo {
    public static void main(String[] args) {
        int a = 5, b = 10;

        System.out.println("Sum: " + a + b);
        System.out.println("Sum: " + (a + b));
        System.out.println(a + b + " is the sum");
    }
}
```

5.Challenge: Accept input strings and concatenate with formatting.

```
import java.util.Scanner;

public class InputConcatenation {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("Enter your first name: ");
        String first = sc.nextLine();

        System.out.print("Enter your last name: ");
        String last = sc.nextLine();

        System.out.print("Enter your city: ");
        String city = sc.nextLine();

        String formatted = "Hello, " + first + " " + last + " from " + city + "!";
        System.out.println(formatted);
    }
}
```

5. StringBuilder

1.Challenge: Reverse a string using StringBuilder.

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

2.Challenge: Append multiple strings using StringBuilder and print.

```
public class AppendStrings {
    public static void main(String[] args) {
        StringBuilder sb = new StringBuilder();
        sb.append("Welcome ").append("to ").append("Java!");
        System.out.println(sb.toString());
    }
}
```

3.Challenge: Replace characters in a string using StringBuilder.

```
public class ReplaceChars {
    public static void main(String[] args) {
        StringBuilder sb = new StringBuilder("A vimala");
        sb.setCharAt(10, '-');
        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("sus");
        sb.insert(1, "hma ");
        System.out.println(sb.toString());
    }
}
```

5.Challenge: Delete part of a string using StringBuilder.

```
public class DeletePart {
    public static void main(String[] args) {
        StringBuilder sb = new StringBuilder("Hello, World!");
        sb.delete(5, 7);
        System.out.println(sb.toString());
    }
}
```

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

```
public class CountVowels {
   public static void main(String[] args) {
      String str = "Vimala";
      int count = 0;
      str = str.toLowerCase();

      for (int i = 0; i < str.length(); i++) {
         char ch = str.charAt(i);
         if ("aeiou".indexOf(ch) != -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 str = "madam";
      String reversed = new StringBuilder(str).reverse().toString();

      if (str.equals(reversed)) {
         System.out.println(str + " is a palindrome.");
      } else {
         System.out.println(str + " 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 str = "Mahesh Babu";

      System.out.println("Upper Case: " + str.toUpperCase());
      System.out.println("Lower Case: " + str.toLowerCase());
   }
}
```

5. 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 & 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(2002, 06, 15);
        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 today = LocalDate.now();
        DateTimeFormatter formatter = DateTimeFormatter.ofPattern("dd-MM-yyyy");

        String formattedDate = today.format(formatter);
        System.out.println("Formatted Date: " + formattedDate);
    }
}
```

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 future = today.plusDays(5);

        System.out.println("Today: " + today);
        System.out.println("After 5 Days: " + future);
    }
}
```

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 num = new BigDecimal("123.456789");
      BigDecimal rounded = num.setScale(2, RoundingMode.HALF_UP);

      System.out.println("Rounded to 2 places: " + rounded);
   }
}
```

8. Flow Control

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

```
public class NumberCheck {
   public static void main(String[] args) {
      int num = 0;

      if (num > 0) {
         System.out.println("Positive");
      } else if (num < 0) {
         System.out.println("Negative");
      } else {
         System.out.println("Zero");
      }
   }
}
```

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

```
public class LargestOfThree {
   public static void main(String[] args) {
      int a = 10, b = 20, c = 30;

      if (a > b) {
         if (a > c) {
            System.out.println("Largest: " + a);
         } else {
            System.out.println("Largest: " + c);
         }
      } else {
         if (b > c) {
            System.out.println("Largest: " + b);
         } else {
            System.out.println("Largest: " + c);
         }
      }
```

```
    }
}
```

3.Challenge: Validate login with username and password.

```
import java.util.Scanner;

public class LoginValidation {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String correctUsername = "admin";
        String correctPassword = "1234";

        System.out.print("Enter username: ");
        String user = sc.nextLine();
        System.out.print("Enter password: ");
        String pass = sc.nextLine();

        if (user.equals(correctUsername) && pass.equals(correctPassword)) {
            System.out.println("Login successful!");
        } else {
            System.out.println("Invalid credentials.");
        }
    }
}
```

4.Challenge: Categorize age groups using if-else ladder.

```
public class AgeGroup {
    public static void main(String[] args) {
        int age = 25;

        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.Challenge: Determine student grade using percentage.

```
public class GradeCalculator {
    public static void main(String[] args) {
        int percentage = 69;
        if (percentage >= 90) {
            System.out.println("Grade: A");
        } else if (percentage >= 75) {
            System.out.println("Grade: B");
        } else if (percentage >= 60) {
            System.out.println("Grade: C");
        } else if (percentage >= 40) {
            System.out.println("Grade: D");
        } else {
            System.out.println("Grade: F (Fail)");
        }
    }
}
```

9. Conditions

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

```
public class EvenAndDivisibleBy5 {
    public static void main(String[] args) {
        int num = 25;

        if (num % 2 == 0 && num % 5 == 0) {
            System.out.println(num + " is even and divisible by 5");
        } else {
            System.out.println(num + " does not meet both conditions");
        }
    }
}
```

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

```
public class TriangleValidator {
    public static void main(String[] args) {
        int angle1 = 60, angle2 = 60, angle3 = 60;

        if (angle1 + angle2 + angle3 == 180) {
            System.out.println("Valid triangle");
        } else {
            System.out.println("Invalid triangle");
        }
    }
}
```

3.Challenge: Check if year is a leap year.

```
public class LeapYearCheck {
    public static void main(String[] args) {
        int year = 2025;

        if ((year % 4 == 0 && year % 100 != 0) || (year % 400 == 0)) {
            System.out.println(year + " is a leap year");
        } else {
            System.out.println(year + " is not a leap year");
        }
    }
}
```

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

```
public class CharacterType {
    public static void main(String[] args) {
        char ch = '_';

        if (Character.isDigit(ch)) {
            System.out.println("Digit");
        } else if (Character.isLetter(ch)) {
            if ("aeiouAEIOU".indexOf(ch) != -1) {
                System.out.println("Vowel");
            } else {
                System.out.println("Consonant");
            }
        } else {
            System.out.println("Special character");
        }
    }
}
```

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

```
public class EligibilityChecker {
    public static void main(String[] args) {
        int age = 22;

        if (age >= 18) {
            System.out.println("Eligible to vote");
        }
        if (age >= 18) {
            System.out.println("Eligible for driving license");
        }
        if (age >= 21) {
            System.out.println("Eligible for government job");
        }
    }
}
```

10. Switch

1.Challenge: 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 first number: ");
        int a = sc.nextInt();
        System.out.print("Enter second number: ");
        int b = sc.nextInt();
        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");
        }
    }
}
```

Output:

2.Challenge: Map number to month name using switch.

```
public class MonthMapper {
    public static void main(String[] args) {
        int month = 4;

        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");
        }
    }
}
```

3.Challenge: Implement a simple menu using switch.

```
public class MenuProgram {
    public static void main(String[] args) {
        int choice = 2;

        System.out.println("1. Start\n2. Settings\n3. Exit");

        switch (choice) {
            case 1: System.out.println("Starting..."); break;
            case 2: System.out.println("Opening settings..."); break;
            case 3: System.out.println("Exiting..."); break;
            default: System.out.println("Invalid choice");
        }
    }
}
```

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

```
public class EnhancedSwitch {
    public static void main(String[] args) {
        String day = "Monday";

        switch (day) {
            case "Monday" -> System.out.println("Start of work week");
            case "Friday" -> System.out.println("End of work week");
            case "Saturday", "Sunday" -> System.out.println("Weekend!");
            default -> System.out.println("Midweek");
        }
    }
}
```

5.Challenge: Implement day of the week based on integer input.

```
public class DayOfWeek {
    public static void main(String[] args) {
        int day = 6;

        switch (day) {
            case 1: System.out.println("Monday"); break;
            case 2: System.out.println("Tuesday"); break;
            case 3: System.out.println("Wednesday"); break;
            case 4: System.out.println("Thursday"); break;
            case 5: System.out.println("Friday"); break;
            case 6: System.out.println("Saturday"); break;
            case 7: System.out.println("Sunday"); break;
            default: System.out.println("Invalid day number");
        }
    }
}
```

11. Loop & Branching

1.Challenge: Print multiplication table for a number.

```
public class MultiplicationTable {
    public static void main(String[] args) {
        int num = 5;
        for (int i = 1; i <= 10; i++) {
            System.out.println(num + " x " + i + " = " + (num * i));
        }
    }
}
```

2.Challenge: Use break and continue in loops.

```
public class BreakContinueDemo {
    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.Challenge: Find factorial of a number.

```
public class FactorialCalc {
    public static void main(String[] args) {
        int num = 5;
        long fact = 1;

        for (int i = 1; i <= num; i++) {
            fact *= i;
        }

        System.out.println("Factorial of " + num + " is " + fact);
    }
}
```

4.Challenge: Print Fibonacci series.

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

        System.out.print("Fibonacci Series: ");
        for (int i = 1; i <= n; i++) {
            System.out.print(a + " ");
            int next = a + b;
            a = b;
            b = next;
        }
    }
}
```

5.Challenge: 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.Challenge: Find the largest and smallest element in an array.

```
public class MinMaxInArray {
    public static void main(String[] args) {
        int[] arr = {12, 5, 8, 20, 3};
        int min = arr[0], max = arr[0];

        for (int num : arr) {
            if (num < min) min = num;
            if (num > max) max = num;
        }

        System.out.println("Minimum: " + min);
        System.out.println("Maximum: " + max);
    }
}
```

2.Challenge: Sort an array in ascending order.

```
import java.util.Arrays;

public class SortArray {
    public static void main(String[] args) {
        int[] arr = {12, 5, 8, 20, 3};

        Arrays.sort(arr);

        System.out.print("Sorted array: ");
        for (int num : arr) {
            System.out.print(num + " ");
        }
    }
}
```

3.Challenge: Calculate average of numbers in an array.

```
public class AverageArray {
    public static void main(String[] args) {
        int[] arr = {10, 20, 30, 40, 50};
        int sum = 0;

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

        double average = (double) sum / arr.length;
        System.out.println("Average: " + average);
    }
}
```

4.Challenge: Count occurrence of an element.

```
public class ElementCount {
    public static void main(String[] args) {
        int[] arr = {1, 2, 3, 2, 4, 2, 5};
        int target = 2;
        int count = 0;

        for (int num : arr) {
            if (num == target) count++;
        }

        System.out.println("Occurrence of " + target + ": " + count);
    }
}
```

5.Challenge: Reverse elements of an array.

```
public class ReverseArray {
    public static void main(String[] args) {
        int[] arr = {10, 20, 30, 40, 50};

        System.out.print("Reversed array: ");
        for (int i = arr.length - 1; i >= 0; i--) {
            System.out.print(arr[i] + " ");
        }
    }
}
```

13. Enum

1.Challenge: Define an enum for days of the week.

```
public class DaysEnumDemo {
    enum Day { MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, SATURDAY, SUNDAY
}

    public static void main(String[] args) {
        Day today = Day.FRIDAY;
        System.out.println("Today is " + today);
    }
}
```

2.Challenge: Use enum in switch case.

```
public class EnumSwitch {
    enum Day { MONDAY, FRIDAY, SUNDAY }

    public static void main(String[] args) {
        Day day = Day.FRIDAY;

        switch (day) {
            case MONDAY:
                System.out.println("Start of the work week!");
                break;
            case FRIDAY:
                System.out.println("Almost weekend!");
                break;
            case SUNDAY:
                System.out.println("Relax, it's Sunday.");
                break;
        }
    }
}
```

3.Challenge: Iterate over enum values.

```
public class EnumIteration {
    enum Color { RED, GREEN, BLUE }

    public static void main(String[] args) {
        for (Color c : Color.values()) {
            System.out.println(c);
        }
    }
}
```

4.Challenge: Assign properties to enum constants.

```
public class EnumWithProperties {
    enum Planet {
        EARTH(5.97), MARS(0.642), JUPITER(1898);

        private double mass;
        Planet(double mass) {
            this.mass = mass;
        }

        public double getMass() {
            return mass;
        }
    }

    public static void main(String[] args) {
        System.out.println("Mass of Earth: " + Planet.EARTH.getMass() + " x10^24 kg");
    }
}
```

5.Challenge: Create an enum to represent traffic light states.

```
{
    enum Signal {
        RED, YELLOW, GREEN
    }

    public static void main(String[] args) {
        Signal signal = Signal.RED;

        switch (signal) {
            case RED:
                System.out.println("STOP");
                break;
            case YELLOW:
                System.out.println("READY");
                break;
            case GREEN:
                System.out.println("GO");
                break;
        }
    }
}
```

–