Java Programs for Class 8 ICSE

May 11, 2026

Contents

1. Introduction to Java

1.1. What is Java?

Java is a popular programming language used to create computer programs. It was developed by Sun Microsystems and is now owned by Oracle. Java allows programmers to write code that can run on many different kinds of computers and devices. This makes Java a **platform-independent** language, which means the same program can work on Windows, Mac, Linux, and even mobile phones.

1.2. Importance of Java in School Curriculum

Java is taught in schools because it helps students understand the basics of programming clearly. It introduces important concepts like classes, objects, and methods, which are the foundation of modern programming. Learning Java prepares students for advanced computer studies and practical applications in real life.

1.3. Features of Java

- **Simple and Easy to Learn:** Java has a straightforward syntax similar to English.
- **Platform Independent:** Write once, run anywhere (WORA).
- **Object-Oriented:** Code is organized into classes and objects.
- **Secure:** Java has built-in security features.
- **Robust:** Java programs are reliable and have fewer errors.

- **Multithreaded:** Can perform many tasks simultaneously.

2. Structure of Java Program

A Java program follows a specific structure which must be understood clearly.

2.1. The `class` Keyword

Every Java program has at least one `class`. A class is like a blueprint for objects. The program starts by defining a class using the keyword `class` followed by the class name.

2.2. The `main()` Method

The `main()` method is the entry point of any Java program. When we run a program, the instructions inside `main()` are executed first. It must always be declared as:

```
public static void main(String[] args) {
    // program code
}
```

2.3. `System.out.println()`

This statement is used to print text or values on the screen. The word `System` is a class, `out` is an object representing the screen, and `println()` is a method that prints the output and moves the cursor to the next line.

2.4. Curly Brackets and Semicolons

Curly brackets `{ }` are used to group statements. The opening bracket `{` marks the start of a block, and the closing bracket `}` marks the end.

Every statement in Java ends with a semicolon `;`, which tells the compiler that this instruction is complete.

3. Rules for Writing Java Programs

3.1. Case Sensitivity

Java is case sensitive. That means `MyVar` and `myvar` are two different identifiers. You must be consistent with the use of uppercase and lowercase letters.

3.2. Naming Rules

- Names must start with a letter (A-Z or a-z), dollar sign `$` or underscore `_`.
- They cannot start with numbers.
- Names cannot contain spaces or special characters like `! % #`.
- Use meaningful names for better clarity.

3.3. File Name and Class Name Rules

The file name and the public class name must be the same. For example, if the class name is `HelloWorld`, the file must be saved as `HelloWorld.java`.

4. Basic Output Programs

Here are some simple Java programs that show how to print messages.

4.1. Print Your Name

```
public class PrintName {
    public static void main(String[] args) {
        System.out.println("John Doe");
    }
}
```

Explanation: This program prints the name "John Doe" on the screen.

Output:

```
John Doe
```

4.2. Print School Name

```
public class SchoolName {
    public static void main(String[] args) {
        System.out.println("ABC Higher Secondary School");
```

```
    }
}
```

Explanation: Prints the school name on the screen.

Output:

```
ABC Higher Secondary School
```

4.3. Print Multiple Lines

```
public class MultipleLines {
    public static void main(String[] args) {
        System.out.println("Welcome to Java programming.");
        System.out.println("This is line 2.");
        System.out.println("This is line 3.");
    }
}
```

Explanation: Prints three lines using three `println()` statements.

Output:

```
Welcome to Java programming.
This is line 2.
This is line 3.
```

4.4. Print a Welcome Message

```
public class WelcomeMessage {
    public static void main(String[] args) {
        System.out.println("Welcome to Class 8 ICSE Java
   Learning!");
    }
}
```

Explanation: Prints a welcome message for Class 8 ICSE students.

Output:

```
Welcome to Class 8 ICSE Java Learning!
```

5. Java Programs Using Variables

Variables store data values. Java supports different types of variables.

5.1. Integer Variables

```
public class IntegerExample {
    public static void main(String[] args) {
        int age = 13;
        System.out.println("My age is " + age);
    }
}
```

Explanation: Declares an integer variable age and prints it.

Output:

```
My age is 13
```

5.2. Decimal Numbers

```
public class DecimalExample {
    public static void main(String[] args) {
        double price = 99.99;
        System.out.println("Price is " + price);
    }
}
```

Explanation: Uses `double` to store decimal numbers.

Output:

```
Price is 99.99
```

5.3. Character Variables

```
public class CharExample {
    public static void main(String[] args) {
        char grade = 'A';
        System.out.println("Grade: " + grade);
    }
}
```

Explanation: Stores a single character using `char` type.

Output:

```
Grade: A
```

5.4. String Variables

```
public class StringExample {
    public static void main(String[] args) {
        String message = "Hello, Java!";
        System.out.println(message);
    }
}
```

Explanation: Strings store multiple characters or text.

Output:

```
Hello, Java!
```

6. Arithmetic Programs in Java

We can perform calculations using arithmetic operators.

6.1. Addition

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

```
    }
}
```

Output:

```
Sum = 30
```

6.2. Subtraction

```
public class Subtraction {
    public static void main(String[] args) {
        int a = 30, b = 15;
        int diff = a - b;
        System.out.println("Difference = " + diff);
    }
}
```

6.3. Multiplication

```
public class Multiplication {
    public static void main(String[] args) {
        int a = 5, b = 6;
        int product = a * b;
        System.out.println("Product = " + product);
    }
}
```

6.4. Division

```
public class Division {
    public static void main(String[] args) {
        int a = 20, b = 4;
        int quotient = a / b;
        System.out.println("Quotient = " + quotient);
    }
}
```

6.5. Modulus (Remainder)

```
public class Modulus {
    public static void main(String[] args) {
        int a = 17, b = 5;
        int remainder = a % b;
        System.out.println("Remainder = " + remainder);
    }
}
```

7. Java Programs Using User Input

7.1. Introduction to Scanner Class

To get input from the user, Java uses the `Scanner` class from `java.util` package.

7.2. Input Integer Values

```
import java.util.Scanner;
public class InputInteger {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter an integer: ");
        int num = sc.nextInt();
        System.out.println("You entered: " + num);
        sc.close();
    }
}
```

Explanation: Program reads an integer typed by the user.

7.3. Input String Values

```
import java.util.Scanner;
public class InputString {
    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);
        sc.close();
    }
}
```

8. Conditional Statement Programs

8.1. `if` Statement

```
public class EvenOdd {
    public static void main(String[] args) {
        int num = 8;
        if(num % 2 == 0) {
            System.out.println(num + " is even.");
        }
    }
}
```

8.2. `if else` Statement

```
public class EvenOdd {
    public static void main(String[] args) {
        int num = 7;
        if(num % 2 == 0) {
            System.out.println(num + " is even.");
        } else {
            System.out.println(num + " is odd.");
        }
    }
}
```

8.3. Nested `if` Basics

```
public class PositiveNegative {
```

```
    public static void main(String[] args) {
        int num = -5;
        if(num >= 0) {
            if(num == 0) {
                System.out.println("Number is zero.");
            } else {
                System.out.println("Number is positive.");
            }
        } else {
            System.out.println("Number is negative.");
        }
    }
}
```

8.4. Largest of Two Numbers

```
public class LargestNumber {
    public static void main(String[] args) {
        int a = 10, b = 20;
        if(a > b) {
            System.out.println(a + " is larger.");
        } else {
            System.out.println(b + " is larger.");
        }
    }
}
```

9. Loop Programs in Java

Loops help to repeat actions.

9.1. `for` Loop

```
public class ForLoop {
    public static void main(String[] args) {
        for(int i = 1; i <= 10; i++) {
            System.out.println(i);
        }
    }
}
```

9.2. `while` Loop

```
public class WhileLoop {
    public static void main(String[] args) {
        int i = 1;
        while(i <= 10) {
            System.out.println(i);
            i++;
        }
    }
}
```

9.3. `do while` Loop

```
public class DoWhileLoop {
    public static void main(String[] args) {
        int i = 1;
        do {
            System.out.println(i);
            i++;
        } while(i <= 10);
    }
}
```

9.4. Multiplication Table

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

9.5. Sum of Natural Numbers

```
public class SumNatural {
    public static void main(String[] args) {
        int sum = 0;
        for(int i=1; i<=10; i++) {
```

```
            sum += i;
        }
        System.out.println("Sum = " + sum);
    }
}
```

10. Pattern Programs

10.1. Star Pattern

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

10.2. Number Pattern

```
public class NumberPattern {
    public static void main(String[] args) {
        for(int i=1; i<=5; i++) {
            for(int j=1; j<=i; j++) {
```

```
                System.out.print(j + " ");
            }
            System.out.println();
        }
    }
}
```

10.3. Simple Triangle Pattern

```
public class TrianglePattern {
    public static void main(String[] args) {
        for(int i=1; i<=5; i++) {
            for(int j=5; j>=i; j--) {
                System.out.print("* ");
            }
            System.out.println();
        }
    }
}
```

11. School-Level Mathematical Programs

11.1. Area of Rectangle

```
public class RectangleArea {
    public static void main(String[] args) {
        int length = 10, breadth = 5;
```

```
        int area = length * breadth;
        System.out.println("Area of rectangle = " + area);
    }
}
```

11.2. Perimeter of Square

```
public class SquarePerimeter {
    public static void main(String[] args) {
        int side = 7;
        int perimeter = 4 * side;
        System.out.println("Perimeter of square = " +
   perimeter);
    }
}
```

11.3. Percentage Calculator

```
public class PercentageCalculator {
    public static void main(String[] args) {
        int totalMarks = 500, obtainedMarks = 450;
        double percentage = (obtainedMarks * 100.0) /
   totalMarks;
        System.out.println("Percentage = " + percentage + "
   %");
    }
}
```

11.4. Simple Interest

```
public class SimpleInterest {
    public static void main(String[] args) {
        double principal = 10000, rate = 5, time = 2;
        double interest = (principal * rate * time) / 100;
        System.out.println("Simple Interest = " + interest)
   ;
    }
}
```

12. Important Viva Questions

- **What is Java?**
  Java is a high-level, object-oriented programming language used to create applications that can run on any platform.
- **What is a class?**
  A class is a blueprint or template from which objects are created in Java.
- **What is the `main()` method?**
  The `main()` method is the starting point of execution of a Java program.
- **What is the Scanner class?**
  Scanner is a class in `java.util` package used to take input from the user.

13. Common Errors in Java

- **Missing semicolon:** Every statement should end with a semicolon. Forgetting it causes errors.
- **Wrong brackets:** Curly brackets must be correctly opened and closed.
- **Variable declaration errors:** Variables should be declared before use and with correct types.
- **Incorrect syntax:** Using keywords wrongly or missing parts of statements causes compile errors.

14. 30 Important Java Programs with Output

Program 1: Print Hello World

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

Explanation: This program prints the famous phrase ”Hello World” to begin Java programming.

Output:

```
Hello World
```

Program 2: Print your Name

```
public class PrintName {
    public static void main(String[] args) {
        System.out.println("Alice");
    }
}
```

Explanation: Prints the name "Alice" on the screen.

Output:

```
Alice
```

Program 3: Addition of Two Numbers

```
public class AddTwoNumbers {
    public static void main(String[] args) {
        int a = 12, b = 8;
        int sum = a + b;
        System.out.println("Sum = " + sum);
    }
}
```

Explanation: Variables a and b store numbers. The sum is calculated and printed.

Output:

```
Sum = 20
```

Program 4: Even or Odd Number Check

```
public class EvenOddCheck {
    public static void main(String[] args) {
        int num = 9;
        if(num % 2 == 0)
            System.out.println(num + " is even.");
        else
            System.out.println(num + " is odd.");
    }
}
```

Explanation: Uses modulus operator to check if the number is even or odd.

Output:

```
9 is odd.
```

Program 5: Print Numbers 1 to 10 using `for` Loop

```
public class NumbersOneToTen {
    public static void main(String[] args) {
        for(int i=1; i<=10; i++) {
            System.out.println(i);
        }
    }
}
```

Explanation: Loop from 1 to 10 prints numbers line by line.

Output:

```
1
2
3
4
5
6
7
8
9
10
```

15. Revision Notes

- Java is platform independent and object-oriented.
- Every program starts with a `class` containing `main()` method.
- Use `System.out.println()` to print output.
- Java is case sensitive; follow naming conventions.
- Variables must be declared with correct type.
- Use semicolons at the end of every statement.
- Use `Scanner` class to get user input.
- Conditional statements (`if`, `if else`) help decision making.
- Loops (`for`, `while`, `do while`) repeat instructions.
- Patterns help in understanding nested loops.
- Common errors: missing semicolon, wrong brackets, variable not declared.

Quick Tips for Exams:

- Always match class name and filename.
- Indent your code properly for readability.
- Understand program logic before coding.
- Practice writing programs by hand as well.