CODE FOR KIDS

JAVA

Learn by doing 10 ready to type game in JAVA

Marco Garagna

Title:*Code for Kids JAVA*
Author: Marco Garagna
Edition: 1st Edition
Published: 2025
ISBN: 9798317176631

Imprint: Independently published

Table of Contents

Introduction

Hey there, future coding superstar!

Welcome to Java Coding for Kids! 🎉 Have you ever wondered how video games, apps, and websites are made? Well, guess what? You’re about to learn one of the coolest secrets—how to code!

In this book, we’ll explore Java, a powerful coding language that helps create awesome things like Minecraft, robots, and even space programs! 🚀 But don’t worry—it’s not as tricky as it sounds. We’ll take it step by step, just like learning to ride a bike. By the end of this book, you’ll be writing your own Java programs!

What you’ll need:

A computer (any kind will do!)

A little curiosity 🧐

A big imagination! 🎨

Are you ready? Let’s dive into the world of Java and start coding some magic! 💻 ✨

1 Setting up for fun.

Learning to code is most fun when you're creating something you can play with! By typing and building these simple games, you're not just learning Java programming concepts—you're also developing problem-solving skills and understanding how code works in real-world applications.

The best way to learn is by doing. As you type each line of code, you'll see how it all comes together to create a working game. You'll learn about variables, loops, and conditional statements in a hands-on way that makes sense. Plus, you can modify the game to make it your own, which is a great way to experiment and learn even more!

So, get started and build your own games. You'll be amazed at how quickly you can create something cool and fun!

How to get started?

If you have already an IDE you can use that or you can type your games online and play without installing extra software.

Here is two simple path you can take to get ready:

1. Online Fun

Open a Web Browser:
Use a browser like Google Chrome, Firefox, or Safari.

Search for "Online Java Compiler":
Type these exact words in the search bar and press Enter.

Choose a Compiler:
Look at the search results and pick a compiler that looks easy to use. Some popular options include:

JDoodle: Known for its simplicity and ease of use.

TutorialsPoint: Offers a comprehensive environment with features like code execution and sharing.

OneCompiler: Provides a robust editor with features like auto-completion and error checking.

2. Use an IDE

If you have the possibility to set up an IDE (Integrated Development Environment) locally with the help of a parent or friend, you should try that! IDEs like IntelliJ IDEA or Eclipse make coding easier and provide powerful tools to help you learn Java.

However, if setting up an IDE feels too complicated right now, don’t worry! You can start coding online using one of the compilers mentioned above. It’s simple, quick, and perfect for beginners.

2 Hello World!

```
import java.util.Scanner;

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

    System.out.print("What is your name? ");
    String name = scanner.nextLine();

    System.out.println("Hello, " + name + "!");

    scanner.close();
    }
}
```

How can you Modify This Game

1. Change the Greeting: Instead of "Hello", use "Hi", "Hey", or any other greeting.
2. Add a Question: Ask the player for their favorite color or animal and include it in the greeting.
3. Use Different Languages: Greet the player in different languages (e.g., Spanish, French).

3 Guessing Games

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

public class GuessTheNumber {
public static void main(String[] args) {
    Random random = new Random();
    int secretNumber = random.nextInt(100) + 1;
    int attempts = 0;
    boolean guessedCorrectly = false;

    Scanner scanner = new Scanner(System.in);
    System.out.println("Welcome to Guess the Number!");
    System.out.println("I'm thinking of a number between 1 and 100.");

    while (!guessedCorrectly) {
        System.out.print("Enter your guess: ");
        int guess = scanner.nextInt();
        attempts++;

        if (guess == secretNumber) {
            guessedCorrectly = true;
        } else if (guess < secretNumber) {
          System.out.println("Too low! Try again.");
        } else {
            System.out.println("Too high! Try again.");
        }
    }

    System.out.println("Congratulations! You guessed the number in " +
       attempts + " attempts.");
    scanner.close();
    }
}
```

How can you Modify This Game?

1. Limit the Number of Attempts – Add a maximum number of guesses (e.g., 10) before the game ends.
2. Add a Hint System – Give hints when the guess is very close (within 5).
3. Change the Number Range – Reduce the range to make it easier (e.g., 1-20 instead of 1-100).
4. Funny Responses – Replace "Too high!" or "Too low!" with fun messages.
5. Multiplayer Mode – Let one player choose a number while another guesses.
6. Play with Sounds (Using Emojis or ASCII Art) – Add emojis or fun ASCII art when they win.

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

public class GuessMyColor {
    public static void main(String[] args) {
        Random random = new Random();
        String[] colors = {"Red", "Blue", "Green", "Yellow", "Purple"};
        String secretColor = colors[random.nextInt(colors.length)];
        int attempts = 0;
        boolean guessedCorrectly = false;

        Scanner scanner = new Scanner(System.in);
        System.out.println("Welcome to Guess My Color!");

        while (!guessedCorrectly && attempts < 5) {
            System.out.print("Guess a color (Red, Blue, Green, Yellow,
                  Purple): ");
            String guess = scanner.nextLine().trim();
            attempts++;

            if (guess.equalsIgnoreCase(secretColor)) {
                guessedCorrectly = true;
            } else {
                System.out.println("Try again!");
            }
        }

        if (guessedCorrectly) {
            System.out.println("Congratulations! You guessed the color
                  in " + attempts + " attempts.");
        } else {
            System.out.println("Sorry, you didn't guess it. The color
                  was " + secretColor + ".");
        }
        scanner.close();
    }
}
```

How can you Modify This Game?

1. **Add More Colors:** Add more colors to the colors array to make the game more challenging.
2. **Change the Number of Attempts:** Adjust the number of attempts allowed.
3. **Use Different Themes:** Instead of colors, use animals, fruits, or any other theme that kids enjoy.
4. Add Hints: Provide hints if the guess is close (e.g., "Your guess is a similar color").

4 Interactive Games

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

  public class RockPaperScissors {
  public static void main(String[] args) {
      Scanner scanner = new Scanner(System.in);
      Random random = new Random();

      String[] choices = {"Rock", "Paper", "Scissors"};

      System.out.println("Welcome to Rock Paper Scissors!");

      while (true) {
          System.out.print("Enter your choice (Rock/Paper/Scissors) or 'quit' to exit: ");
          String playerChoice = scanner.nextLine().toLowerCase();

          if (playerChoice.equals("quit")) {
              break;
          }

          if (!playerChoice.equals("rock") && !playerChoice.equals("paper") && !playerChoice.equals("scissors")) {
            System.out.println("Invalid choice. Please try again.");
              continue;
          }

          int computerIndex = random.nextInt(3);
          String computerChoice = choices[computerIndex];

          System.out.println("Computer chose: " + computerChoice);

          if (playerChoice.equals(computerChoice.toLowerCase())) {
              System.out.println("It's a tie!");
          } else if ((playerChoice.equals("rock") && computerChoice.equals("Scissors")) ||
                      (playerChoice.equals("paper") && computerChoice.equals("Rock")) ||
                      (playerChoice.equals("scissors") && computerChoice.equals("Paper"))) {
              System.out.println("You win!");
          } else {
              System.out.println("Computer wins!");
          }
      }

      System.out.println("Thanks for playing!");
      scanner.close();
  }
}
```

```java
import java.util.Scanner;

public class MathQuiz {
public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);
    int score = 0;

    System.out.println("Welcome to the Math Quiz!");

    for (int i = 0; i < 5; i++) {
        int num1 = (int) (Math.random() * 10) + 1;
        int num2 = (int) (Math.random() * 10) + 1;

        System.out.println("What is " + num1 + " + " + num2 + "?");
        int answer = scanner.nextInt();

        if (answer == num1 + num2) {
            System.out.println("Correct!");
            score++;
        } else {
            System.out.println("Sorry, that's incorrect. The answer is " + (num1 + num2) + ".");
        }
    }

    System.out.println("Quiz finished! Your score: " + score + " out of 5.");
    scanner.close();
 }
}
```

How Kids Can Modify This Game

1. Change the Number of Questions: Adjust the loop to ask more or fewer questions by changing the number in the for loop condition.
2. Modify the Difficulty: Change the range of random numbers by adjusting the values in the Math.random() * 10 part. For example, * 20 would give numbers up to 20.
3. Add Different Operations: Include subtraction, multiplication, or division questions by modifying the arithmetic operation in the question and answer check.
4. Implement a Difficulty Level: Ask the player to choose an easy, medium, or hard difficulty at the start, then adjust the number range accordingly.
5. Add a Time Limit: Implement a timer for each question or for the entire quiz to add an extra challenge.

5 Advance Projects

```java
import java.util.Scanner;

public class Hangman {
private static String[] words = {"java", "programming", "computer", "code",
  "developer"};
private static String word = words[(int) (Math.random() * words.length)];
private static String asterisk = new String(new
  char[word.length()]).replace("\0", "*");
private static int count = 0;

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

    while (count < 7 && asterisk.contains("*")) {
        System.out.println("Guess any letter in the word");
        System.out.println(asterisk);
        String guess = sc.next();
        hang(guess);
    }
    sc.close();
}

public static void hang(String guess) {
    String newasterisk = "";
    for (int i = 0; i < word.length(); i++) {
        if (word.charAt(i) == guess.charAt(0)) {
            newasterisk += guess.charAt(0);
        } else if (asterisk.charAt(i) != '*') {
            newasterisk += word.charAt(i);
        } else {
            newasterisk += "*";
        }
    }

    if (asterisk.equals(newasterisk)) {
        count++;
        System.out.println("Wrong guess, try again");
    } else {
        asterisk = newasterisk;
    }
    if (asterisk.equals(word)) {
        System.out.println("Correct! You win! The word was " + word);
    }
}
}
```

How Can you Modify This Game

1. **Add More Words**: Pick words from your favorite shows, animals, or hobbies!
2. **Give a Friendly Message When They Win or Lose**: Make the ending more exciting with custom messages:
3. **Make the Word Appear in Uppercase**: So it looks cleaner and easier to read.

```java
import java.util.Scanner;

public class ConnectFour {
    private static char[][] board = new char[6][7];
    private static char currentPlayer = 'X';

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        initializeBoard();

        while (true) {
            printBoard();
            System.out.println("Player " + currentPlayer + ", choose a column (1-7): ");
            int column = scanner.nextInt() - 1;

            if (dropPiece(column)) {
                if (checkWin()) {
                    printBoard();
                    System.out.println("Player " + currentPlayer + " wins!");
                    break;
                }
                currentPlayer = (currentPlayer == 'X') ? 'O' : 'X';
            } else {
                System.out.println("Column is full. Try again.");
            }
        }
        scanner.close();
    }

    private static void initializeBoard() {
        for (int i = 0; i < 6; i++) {
            for (int j = 0; j < 7; j++) {
                board[i][j] = '-';
            }
        }
    }

    private static void printBoard() {
        for (int i = 0; i < 6; i++) {
            for (int j = 0; j < 7; j++) {
                System.out.print(board[i][j] + " ");
            }
            System.out.println();
        }
    }

    private static boolean dropPiece(int column) {
        for (int i = 5; i >= 0; i--) {
            if (board[i][column] == '-') {
                board[i][column] = currentPlayer;
                return true;
            }
        }
        return false;
    }

```

```java
  private static boolean checkWin() {
      // Check horizontal locations for win
      for (int i = 0; i < 6; i++) {
          for (int j = 0; j < 4; j++) {
              if (board[i][j] != '-' && board[i][j] == board[i][j + 1] && board[i][j] == board[i][j + 2] && board[i][j] == board[i][j + 3]) {
                  return true;
              }
          }
      }

      // Check vertical locations for win
      for (int i = 0; i < 3; i++) {
          for (int j = 0; j < 7; j++) {
              if (board[i][j] != '-' && board[i][j] == board[i + 1][j] && board[i][j] == board[i + 2][j] && board[i][j] == board[i + 3][j]) {
                  return true;
              }
          }
      }

      // Check positively sloped diagonals
      for (int i = 0; i < 3; i++) {
          for (int j = 0; j < 4; j++) {
              if (board[i][j] != '-' && board[i][j] == board[i + 1][j + 1] && board[i][j] == board[i + 2][j + 2] && board[i][j] == board[i + 3][j + 3]) {
                  return true;
              }
          }
      }

      // Check negatively sloped diagonals
      for (int i = 3; i < 6; i++) {
          for (int j = 0; j < 4; j++) {
              if (board[i][j] != '-' && board[i][j] == board[i - 1][j + 1] && board[i][j] == board[i - 2][j + 2] && board[i][j] == board[i - 3][j + 3]) {
                  return true;
              }
          }
      }

      return false;
  }
}
```

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

public class CatchTheStar {
public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);
    Random random = new Random();

    // Game settings - kids can modify these values
    int gameWidth = 10;       // Width of the game area
    int playerPosition = 5;   // Starting position (middle)
    int score = 0;            // Starting score
    int rounds = 15;          // Number of rounds to play
    char playerChar = '^';    // Character representing the player
    char starChar = '*';      // Character representing stars
    char obstacleChar = 'X';  // Character representing obstacles

    System.out.println("=== CATCH THE STAR ===");
    System.out.println("Move left (L) or right (R) to catch stars (*) and avoid obstacles (X)");
    System.out.println("Stars = 10 points, Obstacles = -5 points");
    System.out.println("Press Enter to start the game!");
    scanner.nextLine();

    // Main game loop
    for (int round = 1; round <= rounds; round++) {
        // Create a falling object (either star or obstacle)
        int objectPosition = random.nextInt(gameWidth) + 1;
        boolean isStar = random.nextBoolean();
        char fallingObject = isStar ? starChar : obstacleChar;

        // Display the falling object
        System.out.println("\nRound " + round + "/" + rounds + " | Score: " + score);
        System.out.println();

        // Show the falling object row
        for (int i = 1; i <= gameWidth; i++) {
            System.out.print(i == objectPosition ? fallingObject : " ");
        }
        System.out.println();

        // Show the player row
        for (int i = 1; i <= gameWidth; i++) {
          System.out.print(i == playerPosition ? playerChar : " ");
        }
        System.out.println();

        // Get player move
        System.out.print("Move (L/R): ");
        String move = scanner.nextLine().toUpperCase();

        // Process player movement
        if (move.equals("L") && playerPosition > 1) {
            playerPosition--;
```

```
           } else if (move.equals("R") && playerPosition < gameWidth) {
              playerPosition++;
          }

          // Check if player caught the object
          if (playerPosition == objectPosition) {
              if (isStar) {
                  System.out.println("You caught a star! +10 points");
                  score += 10;
              } else {
                  System.out.println("Oh no! You hit an obstacle! -5 points");
                  score -= 5;
              }
          } else if (isStar) {
              System.out.println("You missed a star!");
          } else {
              System.out.println("You avoided an obstacle!");
          }
      }

      // Game over - display final score
      System.out.println("\n=== GAME OVER ===");
      System.out.println("Final score: " + score);

      // Rate the performance
      if (score >= 100) {
          System.out.println("Amazing! You're a star catcher champion!");
      } else if (score >= 50) {
          System.out.println("Great job! You're getting good at this!");
      } else if (score >= 0) {
          System.out.println("Not bad! Keep practicing!");
      } else {
          System.out.println("Better luck next time!");
      }

      scanner.close();
  }
}
```

What can you easily modify:

- Change the game width to make it easier or harder
- Change the characters used for the player, stars, and obstacles
- Adjust the scoring system (how many points for stars/obstacles)
- Change the number of rounds
- Modify the final messages based on score
- Add new features (like different types of falling objects)

The game uses only basic Java concepts (variables, loops, conditionals, input/output), making it accessible for beginners. Kids can run this in any Java environment, including online IDEs if they don't have Java installed locally.

6 Make it your own!

As well as helping you tweak these games a little for your own enjoyment and taste, I want to give you the opportunity to take part in a great challenge: to create your own game from scratch! Are you ready?

Try making and testing your own game. If you are happy with the result, send it to me for a review. The best game will become part of the next version of this book, and if your game is selected, you will get the book for free!

Here are some tips to try your luck.

1. The easy way: try to use code elements or pieces of code that you have found in the games you have already made. If you know these code "pieces" already worked, it will be easy to make your game, changing the logic and piecing together different parts from differt game is not illegal. Your imagination is the limit.

2. The hard way: start from scratch, if you are brave enough to try. The sky is the limit.

In both cases, **keep your code short and simple!** The goal is to make it easy to understand, fun to play, and helpful for other kids who want to learn.

Here you can send your game and question:

marcogara24@gmail.com

7 Bugs

Uh-oh! You typed your code, hit the Run button… and nothing happened? Or maybe you got a weird red error message? That's okay—it happens to everyone, even the best programmers in the world!

When a program doesn't work like it should, we call the mistake a bug. But don't worry—bugs aren't real insects crawling through your computer (eww!). A bug is just a little error in your code that stops your program from working properly. Why do bugs happen?

Bugs can sneak into your code when:

- You miss a semicolon (;)
- You type a word wrong (like "System" instead of "Systm")
- You forget a closing bracket (})
- You mix up the order of things

Even experienced programmers make these mistakes. The good news is: bugs can be fixed! That's called debugging, and it's like being a detective for your own code. What should I do if I find a bug?

1. Stay calm. You didn't break the computer.
2. Read the error message. It might tell you what went wrong.
3. Check your code carefully. Look for:
    - Missing semicolons
    - Spelling mistakes
    - Mismatched brackets
4. Compare it to the example. If you're copying code from the book, check every line carefully.

5. Ask for help. A parent, friend, or even an online coding group can lend a hand!

One More Thing...

Don’t get frustrated. Bugs are part of learning and they actually help you become a better coder. Every bug you fix is a high five from your future super-coder self!

So the next time your code doesn't run the way you want, just smile and say:

“Aha! A bug! Time to squash it!”

8 Conclusion

🎉 You did it! 🎉

Wow! Look how much you've learned! You went from knowing nothing about Java to writing your own code. That's a HUGE achievement! 👏

But guess what? This is just the beginning of your coding journey! There's so much more to explore, like making games, apps, and even robots. The best part? You can create anything you dream of. 🚀

If you loved coding, here are some ideas for what to do next:

Try making your own project using what you learned!

Keep practicing by solving fun coding challenges.

Explore more books, videos, and websites about Java.

Share your code with friends and teach them too!

And remember—every expert was once a beginner. Keep coding, keep creating, and keep having fun! 🎮 💡

See you in the world of code! 🖥️ ✨

About the Author

Marco Garagna has always loved technology, solving puzzles, and helping others learn. With a passion for coding and teaching, he decided to write this book to make Java fun and easy for kids.

When he is not coding, Marco enjoys learning about bitcoin, swimming, or building cool projects. He believes that **anyone can code**—all it takes is curiosity and a bit of practice!

Want to share your coding journey? You can reach Marco at:
📧 Email: marcogara24@gmail.com
🌐 Website: https://marcogara.github.io/

Special Acknowledgments

To my son, **Vincent**—thank you for reminding me to be a great father, even when we cannot always be together. You inspire me every day, and this book is a small reflection of the love and wisdom I hope to share with you.

To my beloved brother, **Nicola**—your story reminds me daily of the fragility of life and the limits of our time. Your memory pushes me to be a better person, to cherish every moment, and to always strive for more.

This book is dedicated to both of you. 💙