1. public class Main { public void main(String[] args)
{ System.out.println("Hello, World!"); } }

What error do you get when running this code?

Main method is not static in class Main, please define the main method as:
public static void main(String[] args)

2. public class Main { static void main(String[] args) { System.out.println("Hello, World!"); } }
   What happens when you compile and run this code?

   It is compiled but the error is = Main method is not static in class Main, please define the main method as: public static void main(String[] args)

3. public class Main { public static int main(String[] args) { System.out.println("Hello, World!"); return 0; }

   What error do you encounter? Why is void used in the main method?

   Main method not found in class Main, please define the main method as:
   public static void main(String[] args)
   or a JavaFX application class must extend javafx.application.Application

4. public class Main { public static void main() { System.out.println("Hello, World!"); } }
   What happens when you compile and run this code? Why is String[] args needed?

   Error: Main method not found in class Main, please define the main method as:
   public static void main(String[] args)
   or a JavaFX application class must extend javafx.application.Application

5. public class Main { public static void main(String[] args) { System.out.println("Main method with String[] args"); } public static void main(int[] args) { System.out.println("Overloaded main method with int[] args"); } }

   Can you have multiple main methods? What do you observe?

   No, Main method with String[] args

6. public class Main { public static void main(String[] args) { int x = y + 10; System.out.println(x); } }
   What error occurs? Why must variables be declared?
   java:3: error: cannot find symbol int x = y + 10;
   symbol: variable y
   location: class Main

7. public class Main { public static void main(String[] args) { int x = "Hello"; System.out.println(x); } }
   What compilation error do you see? Why does Java enforce type safety?

   error: incompatible types: String cannot be converted to int
   int x = "Hello";

8. public class Main { public static void main(String[] args) { System.out.println("Hello, World!" } }
   What syntax errors are present? How do they affect compilation?

   error: ')' expected
   System.out.println("Hello, World!"

9. public class Main { public static void main(String[] args) { int class = 10; System.out.println(class); } }
   What error occurs? Why can't reserved keywords be used as identifiers?
   They are having already predefined meaning which is already known to the compiler of Java.

10. public class Main { public void display() { System.out.println("No parameters"); } public void display(int num) { System.out.println("With parameter: " + num); } public static void main(String[] args) { display(); display(5); } }
    What happens when you compile and run this code? Is method overloading allowed?
    error: non-static method display() cannot be referenced from a static context
    display();
    ^
    Main.java:10: error: non-static method display(int) cannot be referenced from a static context
    display(5);
    ^
    2 errors
11. public class Main { public static void main(String[] args) { int[] arr = {1, 2, 3}; System.out.println(arr[5]); } }
    What runtime exception do you encounter? Why does it occur?

    Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: Index 5 out of bounds for length 3
    at Main.main(Main.java:4)

    Since array is of size 3 elements and we are trying to access element no. 5, That's why we are facing this error.

12. public class Main { public static void main(String[] args) { while (true) { System.out.println("Infinite Loop"); } } }
    What happens when you run this code? How can you avoid infinite loops?

    Code will run into infinite loop, we can stop by doing condition as false.

13. public class Main { public static void main(String[] args) { String str = null; System.out.println(str.length()); } }
    What exception is thrown? Why does it occur?
    Exception in thread "main" java.lang.NullPointerException: Cannot invoke "String.length()" because "<local1>" is null
        at Main.main(Main.java:4)

    length is null and we are trying to get its length, so runtime error.

14. public class Main { public static void main(String[] args) { double num = "Hello"; System.out.println(num); } }
    What compilation error occurs? Why does Java enforce data type constraints?
    incompatible types: String cannot be converted to double
        double num = "Hello";

15. public class Main { public static void main(String[] args) { int num1 = 10; double num2 = 5.5; int result = num1 + num2; System.out.println(result); } }
    What error occurs when compiling this code? How should you handle different data types in operations?
    incompatible types: possible lossy conversion from double to int
        int result = num1 + num2;

    We can do simply typecasting by doing "int(5.5)"

16. public class Main { public static void main(String[] args) { int num = 10; double result = num / 4; System.out.println(result); } }
    What is the result of this operation? Is the output what you expected?
    2.0 (answer is in float format.)

17. public class Main { public static void main(String[] args) { int a = 10; int b = 5; int result = a ** b; System.out.println(result); } }
    What compilation error occurs? Why is the ** operator not valid in Java?
    error: illegal start of expression
        int result = a ** b;

18. public class Main { public static void main(String[] args) { int a = 10; int b = 5; int result = a + b * 2; System.out.println(result); } }
    What is the output of this code? How does operator precedence affect the result?
    20
    Operator uses the BODMAS method for precedence.

19. public class Main { public static void main(String[] args) { int a = 10; int b = 0; int result = a / b; System.out.println(result); } }
    What runtime exception is thrown? Why does division by zero cause an issue in Java?
    java.lang.ArithmeticException: / by zero
        at Main.main(Main.java:5)

    We can't divide any number by 0

20. public class Main { public static void main(String[] args) { System.out.println("Hello, World") } }
    What syntax error occurs? How does the missing semicolon affect compilation?
    error: ';' expected
        System.out.println("Hello, World")

21. public class Main { public static void main(String[] args) { System.out.println("Hello, World!"); // Missing closing brace here }
    What does the compiler say about mismatched braces?
    Code is executed by { _ }

22. public class Main { public static void main(String[] args) { static void displayMessage() { System.out.println("Message"); } } }
    What syntax error occurs? Can a method be declared inside another method?
    error: illegal start of expression
        static void displayMessage() {
        ^
    Main.java:7: error: class, interface, enum, or record expected
    }

23. public class Confusion { public static void main(String[] args) { int value = 2; switch(value) { case 1: System.out.println("Value is 1"); case 2: System.out.println("Value is 2"); case 3: System.out.println("Value is 3"); default: System.out.println("Default case"); } } }
    Error to Investigate: Why does the default case print after "Value is 2"? How can you prevent the program from executing the default case?

    Value is 2
    Value is 3
    Default case
    We can prevent it by using break statement.

24. public class MissingBreakCase { public static void main(String[] args) { int level = 1; switch(level) { case 1: System.out.println("Level 1"); case 2: System.out.println("Level 2"); case 3: System.out.println("Level 3"); default: System.out.println("Unknown level"); } } }
    Error to Investigate: When level is 1, why does it print "Level 1", "Level 2", "Level 3", and "Unknown level"? What is the role of the break statement in this situation?
    Level 1
    Level 2
    Level 3
    Unknown level

25. public class Switch { public static void main(String[] args) { double score = 85.0; switch(score) { case 100: System.out.println("Perfect score!"); break; case 85: System.out.println("Great job!"); break; default: System.out.println("Keep trying!"); } } }

    Error to Investigate: Why does this code not compile? What does the error tell you about the types allowed in switch expressions? How can you modify the code to make it work?

    error: patterns in switch statements are a preview feature and are disabled by default.
        switch(score) {
            ^
      (use --enable-preview to enable patterns in switch statements)
    Switch.java:5: error: constant label of type int is not compatible with switch selector type double
            case 100:
                ^
    Switch.java:8: error: constant label of type int is not compatible with switch selector type double
            case 85:
                ^

26. public class Switch { public static void main(String[] args) { int number = 5; switch(number) { case 5: System.out.println("Number is 5"); break; case 5: System.out.println("This is another case 5"); break; default: System.out.println("This is the default case"); } } }

Error to Investigate: Why does the compiler complain about duplicate case labels? What happens when you have two identical case labels in the same switch block?

error: duplicate case label
case 5: