Snippet 1:

public class Main {

public void main(String[] args) {

System.out.println("Hello, World!");

}

}

Q1 What error do you get when running this code?

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

The error happens because, in Java, the main method needs to be static so the program can run without creating an object of the classs

Static = direct Access = Java can run the method immediately.

Non-static = needs an Object = Java must create an object first, which It cannot do before the program starts.

// write code :-

public class Main {

public static void main(String[] args) {

System.out.println("Hello, World!");

}

}

Output = Hello, World!

Snippet 2:

public class Main {

static void main(String[] args) {

System.out.println("Hello, World!");

}

}

Q2 What happens when you compile and run this code?

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

The error happen because , the main method needs to be public the program can run anywhere

public = allows jvm to access the method from anywhere.

// write code :-

public class Main {

public static void main(String[] args) {

System.out.println("Hello, World!");

}

}

Snippet 3:

public class Main {

public static int main(String[] args) {

System.out.println("Hello, World!");

return 0;

}

}

Q3 What error do you encounter? Why is void used in the main method?

Error :- Main method must return a value of type void in class Main, please define the main method as: public static void main(String[] args)

Why is void used in the main :-

No return value need: The main method starts the program and doesn’t need to return any value

JVM requirement : The Java Virtual Machine (JVM) looks for main with void to run the program

Avoids errors: Using any other return type like int or String causes an error, and the program won’t run

Clear and Simple/ Simple Execution: Void means "no return value," making the code easier to understand

// write code :-

public class Main {

public static void main(String[] args) {

System.out.println("Hello, World!");

return 0;

}

}

Snippet 4:

public class Main {

public static void main() {

System.out.println("Hello, World!");

}

}

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

Q4 What happens when you compile and run this code?

**Compilation:** The code compiles without errors because public static void main() is a valid method in Java.

**Execution / Run Time Error :** When you try to run the program using java Main, you’ll get an error

Why is String[] args needed?

// write code :-

public class Main {

public static void main(String[] args) {

System.out.println("Hello, World!");

}

}

Output = Hello, World!

Snippet 6:

public class Main {

public static void main(String[] args) {

int x = y + 10;

System.out.println(x);

}

}

Error :-

Main.java:3: error: cannot find symbol

int x = y + 10;

^

symbol: variable y , location: class 1 error

Error**:** cannot find variable y value and type

// write code :-

public class Main {

public static void main(String[] args) {

int y = 75; // Variable y is declared and initialized

int x = y + 10;

System.out.println(x);

}

}

Output = 85

Snippet 7 :

public class Main {

public static void main(String[] args) {

int x = "Hello";

System.out.println(x);

}

}

Error :-

error: incompatible types: String cannot be converted to int

int x = "Hello";

What compilation error do you see?

Ans = incompatible types

Write code :-

public class Main {

public static void main(String[] args) {

String x = "Hello"; // assign string data type

System.out.println(x);

}

}

Output = Hello

Why does Java enforce type safety ?

It ensures that variables store only the correct type of data

Better Performance: It helps Java use memory efficiently.

Snippet 8:

public class Main {

public static void main(String[] args) {

System.out.println("Hello, World!"

}

}

Q What syntax errors are present?

Error : - ')' expected

Q How do they affect compilation?

Compilation Failure: The code will not run until you add the missing parenthesis ) and semicolon ;

// Write code

public class Main {

public static void main(String[] args) {

System.out.println("Hello, World!");

}

}

//Snippet 9:

public class Main {

public static void main(String[] args) {

int class = 10;

System.out.println(class);

}

}

Q What error occurs?

C:\Users\91939\Desktop\CDAC\assignment 3>javac Main.java

Main.java:7: error: not a statement

int class = 10;

^

Main.java:7: error: ';' expected

int class = 10;

^

Main.java:7: error: <identifier> expected

int class = 10;

^

Main.java:8: error: <identifier> expected

System.out.println(class);

^

Main.java:8: error: illegal start of type

System.out.println(class);

^

Main.java:8: error: <identifier> expected

System.out.println(class);

^

Main.java:10: error: reached end of file while parsing

}

^

7 errors

Why can't reserved keywords be used as identifiers?

// write code

public class Main {

public static void main(String[] args) {

int a = 10; // change variable name

System.out.println(a);

}

}

Output = 10

//Snippet 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);

}

}

Q What happens when you compile and run this code?

Error :-

Main.java:12: error: non-static method display() cannot be referenced from a static context

display();

^

Main.java:13: error: non-static method display(int) cannot be referenced from a static context

display(5);

2 errors

//Snippet 10:

Q Is method overloading allowed?

overloading method is allowed and works correctly in Java

// write code

public class Main1{

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) {

Main1 obj = new Main1(); // create object

obj.display();

obj.display(5);

}

}

//Snippet 11:

public class Main {

public static void main(String[] args) {

int[] arr = {1, 2, 3};

System.out.println(arr[5]);

}

}

Q What runtime exception do you encounter? Why does it occur?

Error ;- C:\Users\91939\Desktop\CDAC\assignment 3>java Main

Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: Index 5 out of bounds for length 3

at Main.main(Main.java:4)

Runtime Exception: ArrayIndexOutOfBoundsException

Reason: This happens because the code tries to use index 5, but the array only has elements at index 0, 1, and 2. Index 5 is outside the array's limit, so the error occurs.

// write code

public class Main {

public static void main(String[] args) {

int[] arr = {1, 2, 3};

System.out.println(arr[2]);

}

}

Output = 3 = output 3 because index 2 value is 3

//Snippet 12:

public class Main {

public static void main(String[] args) {

while (true) {

System.out.println("Infinite Loop");

}

}

}

Q What happens when you run this code?

The program prints "Infinite Loop" repeatedly without stopping.

Q How can you avoid infinite loops?

Using condition and break statement

//code :-

public class Main {

public static void main(String[] args) {

while (true) {

System.out.println("Infinite Loop");

break;//using break statement stop infinite loop

}

}

}

// using condition

public class Main {

public static void main(String[] args) {

int i = 0;//intialization

while (i <= 5)//condition {

System.out.println("Infinite Loop");

i++; // means i = i + 1;

}

}

}

//Snippet 13:

public class Main {

public static void main(String[] args) {

String str = null;

// null means no any value and str value always written in " "

System.out.println(str.length());

}

}

Q What exception is thrown? Why does it occur?

Error:- C:\Users\91939\Desktop\CDAC\assignment 3>java Main

Exception in thread "main" java.lang.NullPointerException

at Main.main(Main.java:4)

public class Main {

public static void main(String[] args) {

String str = "karan";

System.out.println(str.length());

}

}

Output :- 5

//Snippet 14:

public class Main {

public static void main(String[] args) {

double num =”Hello”;// incompatable data type

System.out.println(num);

}

}

Q What compilation error occurs? Why does Java enforce data type constraints?

Error :-

C:\Users\91939\Desktop\CDAC\assignment 3>javac Main.java

Main.java:6: error: incompatible types: String cannot be converted to double

double num = "Hello";

^1 error

Code :-

public class Main {

public static void main(String[] args) {

String num = "Hello"; // change datatype double to String

System.out.println(num);

}

}

Output = Hello

Snippet 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);

}

}

Q What error occurs when compiling this code? How should you handle different data types in operations?

Error :-

Main.java:11: error: incompatible types: possible lossy conversion from double to int

int result = num1 + num2;

^1 error

Write code :-

public class Main {

public static void main(String[] args) {

int num1 = 10;

double num2 = 5.5;

int result = num1 +(int) num2; // type casting

System.out.println(result);

}

}

Output :- 15

// Snippet 16:

public class Main {

public static void main(String[] args) {

int num = 10;

double result = num / 4;

System.out.println(result);

}

}

Output :- 2.0

Q What is the result of this operation? Is the output what you expected?

num is int with the valaue 10

4 also int

when divide two integer in java the result also an integer = 10/4 = 2 not 2.5

This integer result (2) is then assigned to the double result, which converts 2 to 2.0.

Snippet 17:

public class Main {

public static void main(String[] args) {

int a = 10;

int b = 5;

int result = a \*\* b;

System.out.println(result);

}

}

Q What compilation error occurs? Why is the \*\* operator not valid in Java?

Main.java:5: error: illegal start of expression

int result = a \*\* b;

^

1 error

public class Main{

public static void main(String[] args) {

int a = 10;

int b = 5;

// use Math.pow(base,exponent)

int result = (int)(Math.pow(a,b));

System.out.println(result);

}

}

Output :- 100,000

Snippet 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);

}

}

Q What is the output of this code? How does operator precedence affect the result?

Output :- 20 Because

a + b \* 2

10 + 5 \*2

10 + 10 = 20

//Snippet 19:

public class Main {

public static void main(String[] args) {

int a = 10;

int b = 0;

int result = a / b;

System.out.println(result);

}

}

Q What runtime exception is thrown? Why does division by zero cause an issue in Java

Exception in thread "main" java.lang.ArithmeticException: / by zero

at Main.main(Main.java:5)

//divide by zero not allowed in java

code ;-

public class Main {

public static void main(String[] args) {

int a = 10;

int b = 3;// change value 0 to 3

int result = a / b;

System.out.println(result);

}

}

Output :- 3

//Snippet 20:

public class Main {

public static void main(String[] args) {

System.out.println("Hello, World")

}

}

**Q What syntax error occurs?**

Error :- Main.java:3: error: ';' expected

System.out.println("Hello, World")

**Q How does the missing semicolon affect compilation?**

Java uses semicolons to mark the end of each statement. Without it, the compiler can’t understand where the statement ends.

A missing semicolon causes a **;** expected error.

Write code :-

public class Main {

public static void main(String[] args) {

System.out.println("Hello, World")

}

}

Snippet 21:

public class Main {

public static void main(String[] args) {

System.out.println("Hello, World!");

// Missing closing brace here

}

**Q What does the compiler say about mismatched braces?**

Error: - reached end of file while parsing

}

^

Write code : –

public class Main {

public static void main(String[] args) {

System.out.println("Hello, World!");

// Missing closing brace here

}

}

Output :- Hello, World!

//Snippet 22:

public class Main {

public static void main(String[] args) {

static void displayMessage() {

System.out.println("Message");

}

}

}

**Q What syntax error occurs? Can a method be declared inside another method?**

**Main.java:4: error: illegal start of expression**

**static void displayMessage() {**

**^**

**Main.java:8: error: class, interface, or enum expected**

**}**

// Write code

public class Main {

static void displayMessage() {

System.out.println("Message");

}

public static void main(String[] args) {

displayMessage();

}

}

//Snippet 23:

public class Main {

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");

}

}

}

Output :- Value is 1 , Value is 2 , Default case

value is 2, so case 2 is matched.

System.out.println("Value is 2"); is executed.

Since there is no break, the program continues to:

case 3 → prints "Value is 3"

default → prints "Default case"

//Snippet 24:

public class Main {

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");

}

}

}

Output :-

C:\Users\91939\Desktop\CDAC\assignment 3>java Main

Level 1

Level 2

Level 3

Unknown level

**If you don't use the break statement in a switch, all the next cases will run after the first match.**

Snippet 25 :

public class Main{

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 :- Main.java:4: error: incompatible types: possible lossy conversion from double to int

switch(marks) {

error because the switch statement does not support double and float values. In Java.

//Snippet 26:

public class Main {

public static void main(String[] args) {

int number = 5;

switch(number) {

case 5:

System.out.println("Number is 5");

break;

case 5:// DUPLICATE case lable not allowed

System.out.println("This is another case 5");

break;

default:

System.out.println("This is the default case");

}

}

}

error: duplicate case label

case 5:

**changing the second case lable from case 5 to case 6**

**then error not found**