Snippet 1:

```
public class Test1 {
 public void main(String[] args) {
 System.out.println("Hello, World!");
 }
}
```

Error:

```
Test1.java:1: error: class Main is public, should be declared in a file named Main.java
public class Main {
```

que.What error do you get when running this code?

Ans= compiler shows an error public(access modifier) declared in the main file.and static is missing in that static is required for the main method.

correct code:

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

Snippet 2:

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

- **What happens when you compile and run this code?**

1.Main method missing(public)error:In that main method is missing we must declare the main method as public static void main then our code will run.

2.Main method error:its showing main method can not declare as file

Correct code:

```
public class Test2 {
public static void main(String[] args) {
 System.out.println("Hello, World!");
 }
}
```

---

snnipt 3:

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

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

1.Error: Main method must return a value of type void in class snnipt3, please

define the main method as: in that void is missing thats why showing this error.

jvm need void because it is entry point of the program.

2.error: <identifier> expected

```
 public static int void main(String[] args) {
```

in the public static void int main is this given in the code but int is not correct in that that's why it showing an error.when we remove int code run properly.

3. incompatible types: unexpected return value

return 0; we can not used return 0 in java directly we need System.exit(0);

correct code:

```
public class snnipt3 {
 public static  void main(String[] args) {
 System.out.println("Hello, World!"); }}
```

snnipt4 :

```
public class snnipt4 {
 public static void main() {
 System.out.println("Hello, World!");
 }
}
```

What happens when you compile and run this code? Why is String[] args needed?

```
C:\Users\KALYANI\OneDrive\Desktop\java1>java snnipt4
Error: Main method not found in class snnipt4, please define the main method as:
   public static void main(String[] args)
or a JavaFX application class must extend javafx.application.Application.
```

Main method is missing in the code we reqired the main main method (String args[]) is missing in that that's why showing this error.

correct code:

```
public class snnipt4 {
 public static void main(String args[]) {
 System.out.println("Hello, World!");
 }
}
```

snnipt5:

```
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?

Yes, Java allows method overloading, so you can define multiple methods with the same name but different parameter types. However, only the `main(String[] args)` method is recognized as the program entry point.

The JVM **only execute** the `main(String[] args)` method.

---

snnipt6:

```
public class snnipt6

{

 public static void main(String[] args) {

 int x = y + 10; //error variable not define

 System.out.println(x);

 }

}
```

- **What error occurs? Why must variables be declared?**

```
snnipt6.java:4: error: cannot find symbol

 int x = y + 10;

  symbol:   variable y
```

variable is not declare in that we must declare a variable y in that.

Correct code :

```
public class snnipt6 {

 public static void main(String[] args) {

int y=12;

 int x = y + 10;

 System.out.println(x);

 }}
```

snnipt7 :

```
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?**

```
snnipt7.java:3: error: incompatible types: String cannot be converted to int
 int x = "Hello";
```

it's showing compilation error because you are trying to assign a String ("Hello") to an int variable (x), which is not allowed.

correct code:

```
public class snnipt7 {
 public static void main(String[] args) {
String x = "Hello";
 System.out.println(x);
 }
}
```

Snippet 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?**

```
snnipt8.java:3: error: ')' expected
 System.out.println("Hello, World!"
```

syntax error due to a missing closing parenthesis) in the System.out.println() statement.

Correct code:

```
public class snnipt8 {

 public static void main(String[] args) {

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

 }

}
```

snnipt 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?**

```
snnipt9.java:3: error: not a statement

 int class = 10; snnipt9.java:3: error: ';' expected

 int class = 10; snnipt9.java:3: error: <identifier> expected

 int class = 10;

snnipt9.java:4: error: <identifier> expected

 System.out.println(class);

            ^

snnipt9.java:4: error: illegal start of type

 System.out.println(class);

             ^

snnipt9.java:4: error: <identifier> expected

 System.out.println(class);

                ^

snnipt9.java:6: error: reached end of file while parsing

}

 ^

7 errors
```

class is a reserved keyword in Java, meaning it has a special function (it is used to define classes).

Java does not allow keywords to be used as variable names.

correct code:

```
public class snnipt9 {
 public static void main(String[] args) {
 int a = 10;
 System.out.println(a);
 }
}
```

snnipt10

```
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(); //error
 display(5); //error
 }
}
```

```
snnipt10.java:9: error: non-static method display() cannot be referenced from a static context
 display();
snnipt10.java:10: error: non-static method display(int) cannot be referenced from a static context
 display(5)
2 errors
```

compiler error because you are trying to call the display() method without creating an instance of the class. Since display() is a non-static method, it cannot be called directly inside main(), which is a static method.

correct code:

```
public class snnipt10 {
 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) {
snnipt10 obj=new snnipt10();
 obj.display();
 obj.display(5);
 }
}
```

---

snnipt11

```
public class snnipt11 {
 public static void main(String[] args) {
 int[] arr = {1, 2, 3};
 System.out.println(arr[5];//error
 }
}
```

- **What exception is thrown? Why does it occur**

```
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: Index 5 out of bounds for length 3
	at snnipt11.main(snnipt11.java:4)
```

1.code will compile successfully but in run time its showing error declared an array of size 3 (arr = {1, 2, 3), which means valid indices are 0 to 2.

2.Accessing arr[5] is invalid because index 5 is out of range.

Correct code:

```
public class snnipt11 {
 public static void main(String[] args) {
 int[] arr = {1, 2, 3};
 System.out.println(arr[2];
 }
}
```

snnipt 12

```
public class snnipt12{
 public static void main(String[] args) {
 while (true) {
 System.out.println("Infinite Loop");
 }
 }
}
```

que1.What happens when you run this code?

ans=its run successfully infinite times infinite loop occur

que2.How can you avoid infinite loops?

ans=give the proper initialization,declaration and increment

correct code:

```
public class snnipt12{
 public static void main(String[] args) {
```

```
int i=0;
 while (i<5){
 System.out.println(i);
i++;
 }
 }
}
```

snnipt13

```
public class snnipt13 {
 public static void main(String[] args) {
 String str = null;
 System.out.println(str.length()); //error
 }
}
```

Error:

- **What exception is thrown? Why does it occur?**

```
Exception in thread "main" java.lang.NullPointerException
at snnipt13.main(snnipt13.java:4)
```

it compile successfully but showing runtime error str is null, calling str.length() will cause an exception because there is no actual string object to get the length from.

Correct code:

```
public class snnipt13 {
 public static void main(String[] args) {
 String str = "null"; //add double cot in that
 System.out.println(str.length());
 }
}
```

snnipt14

```
public class snnipt14 {
 public static void main(String[] args) {
 double num = "Hello"; //errorr
 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";
```

datatype is double we assigning string value in that so can change the datatype double replace with string.

Correct code

```
public class snnipt14 {
 public static void main(String[] args) {
String num = "Hello";//datatype double replace with string
 System.out.println(num);
 }
}
```

Snnipt15

```
public class snnipt15 {
 public static void main(String[] args) {
 int num1 = 10;
 double num2 = 5.5;
 int result = num1 + num2; //error
 System.out.println(result);
 }
}
```

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

error: incompatible types: possible lossy conversion from double to int

int result = num1 + num2;

type mismatch error int can not coveted into double.

correct code:

```
public class snnipt15 {
 public static void main(String[] args) {
 int num1 = 10;
 double num2 = 5.5;
 double result = num1 + num2; //change int to double
 System.out.println(result);
 }
}
```

snnipt16

```
public class snnipt16 {
 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?**

program run sucessfully no error output is 2.0

snnipt17

```
public class snnipt17 {
 public static void main(String[] args) {
 int a = 10;
 int b = 5;
 int result = a ** b; //error
 System.out.println(result);
 }
}
```

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

```
snnipt17.java:5: error: illegal start of expression
 int result = a ** b;
```

incorrect because ** is not a valid operator in Java.

Java does not have a built-in exponentiation operator.

Correct code:

```
public class Snippet17 {
    public static void main(String[] args) {
        int a = 10;
        int b = 5;
        double result = Math.pow(a, b); //Use Math.pow() for exponentiation
        System.out.println(result); // Output: 100000.0
    }
}
```

snnipt18

```
public class  snnipt18{
 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?**

Ans=Outout is 20

snnipt19

```
public class Main {
 public static void main(String[] args) {
 int a = 10;
 int b = 0;
 int result = a / b; //error
 System.out.println(result);
 }
}
```

- **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 snnipt19.main(snnipt19.java:5)
```

In Java division by zero is not allowed for integers.

a / b → 10 / 0 causes a runtime error (ArithmeticException).

Java does not allow division by zero for int types, but for double types, it results in Infinity or NaN.

corrrect code

```
public class Main {
 public static void main(String[] args) {
 int a = 10;
 int b = 10;
 int result = a / b;
 System.out.println(result);
 }
}
```

snnipt20

```
public class  snnipt20 {
 public static void main(String[] args) {
 System.out.println("Hello, World") //error semocolon missing
 }
}
```

- **What syntax error occurs? How does the missing semicolon affect compilation?**

```
snnipt20.java:3: error: ';' expected
 System.out.println("Hello, World")
                    ^
```

correct code

```
public class  snnipt20 {
 public static void main(String[] args) {
 System.out.println("Hello, World") ;
 }
}
```

snnipt 21

```
public class snnipt21
{
 public static void main(String[] args) {
 System.out.println("Hello, World!");
 // Missing closing brace here
}
```

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

snnipt21.java:5: error: reached end of file while parsing} (miising } brace)

Correct code

```
public class snnipt21
{
 public static void main(String[] args) {
 System.out.println("Hello, World!");
}
}
```

snnipt22

```
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?**

```
snnipt22.java:3: error: illegal start of expression
 static void displayMessage() {
snnipt22.java:7: error: class, interface, or enum expected
}
```

snnipt23

```
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?

Ans=code is correct output is Value is 2 Value is 3 Default case

Why does the default case print after "Value is 2"?

Ans=because case break is not give after case2 that's by default case execute

How can you prevent the program from executing the default case?

ans=given the breck after in each case

```
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");

breck;

 case 3:

 System.out.println("Value is 3");
 default:
 System.out.println("Default case");
 }
 }
}
```

snnipt24

```
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");
 }
 }
}
```

When level is 1, why does it print "Level 1", "Level 2", "Level 3", and "Unknown level"?

ans=because in that code break starement is not given thats why execte all cases

What is the role of the break statement in this situation?

ans=break statement break the each case .

snnipt25

```
public class Switch {
 public static void main(String[] args) {
 double score = 85.0; //error double not support switch
 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?

```
Switch.java:5: error: incompatible types: possible lossy conversion from double to int
 switch(score) {
```

Switch in java does not suppport double only support int ,char,byte,short.

correct code

```
public class Switch {
 public static void main(String[] args) {
int score = 85.0; //error double not support switch
 switch(score) {
 case 100:
 System.out.println("Perfect score!");
 break;
 case 85:
```

```
 System.out.println("Great job!");
 break;
 default:
 System.out.println("Keep trying!");
 }
 }
}//output =Great job
```

^

^