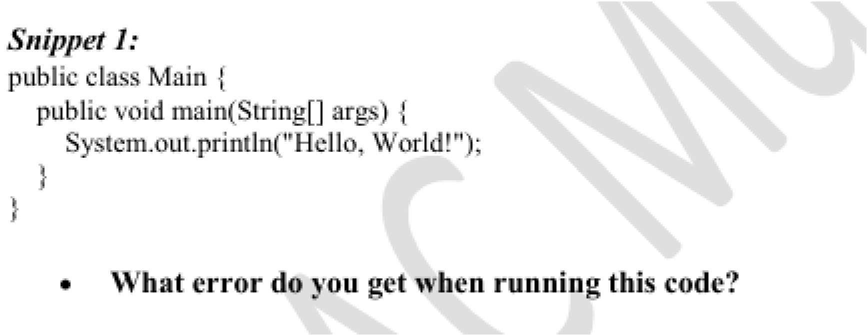
Assignment 2

Section 1



Snippet 1:

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

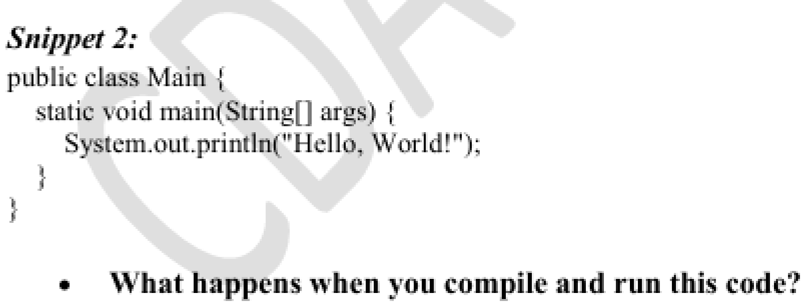
- **What error do you get when running this code?**

Ans:-

ERROR!

Error: Main method is not static in class Main, please define the main method as:

public static void main(String[] args)



Snippet 2:

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

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

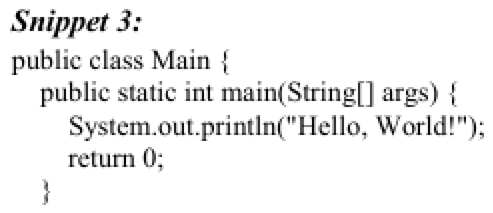
Ans:-

ERROR!

Error: Main method not found in class Main, please define the main method as:

public static void main(String[] args)

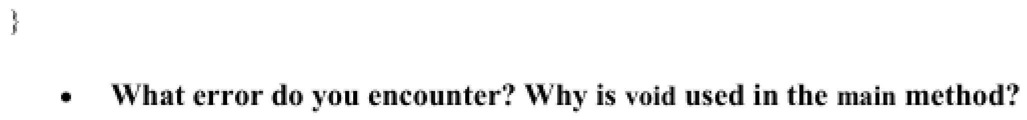
or a JavaFX application class must extend javafx.application.Application



Snippet 3:

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

[redacted]



```
}
```

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

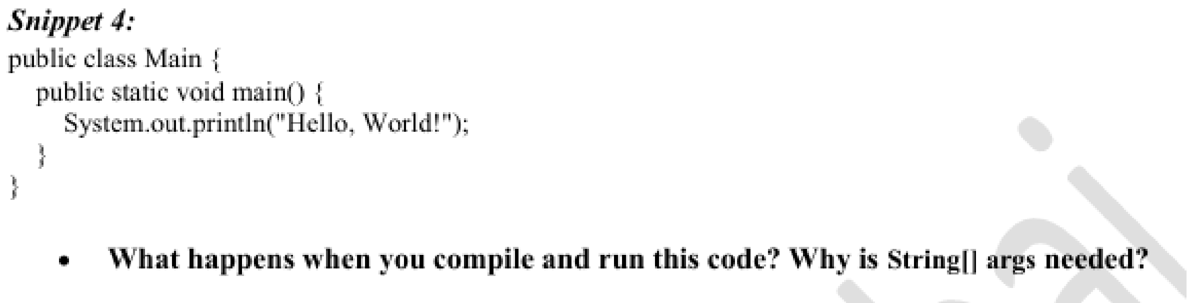
Ans

ERROR!

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)



Snippet 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?**

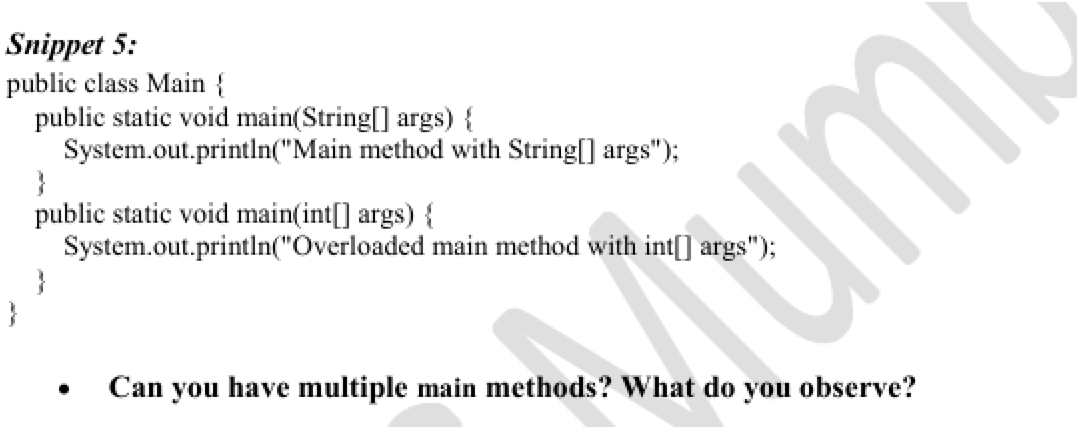
Ans

ERROR!

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



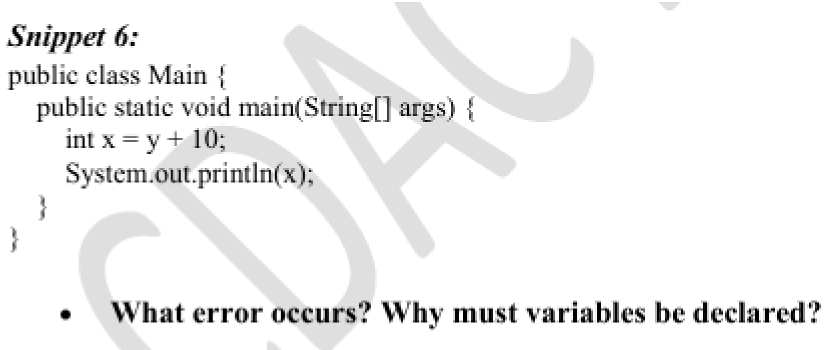
Snippet 5:

```
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?**

Ans

Main method with String[] args



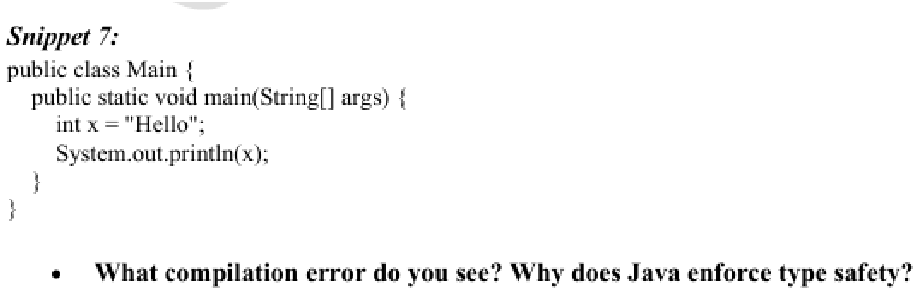
Snippet 6:

```
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?**

Ans

```
     int x = y + 10;

             ^

  symbol:   variable y

  location: class Main

1 error
```



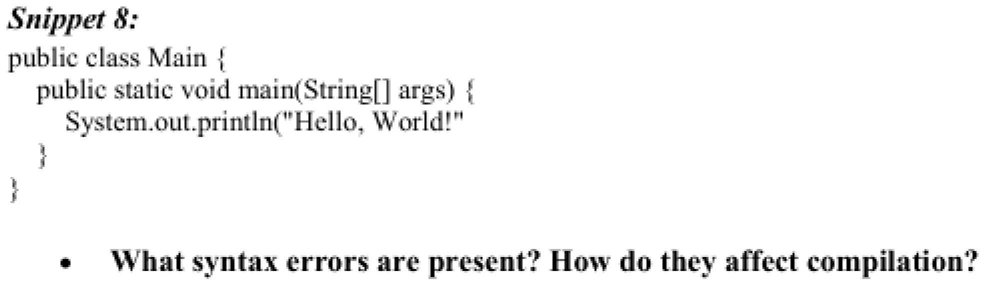
Snippet 7:

```
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?**

Ans

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



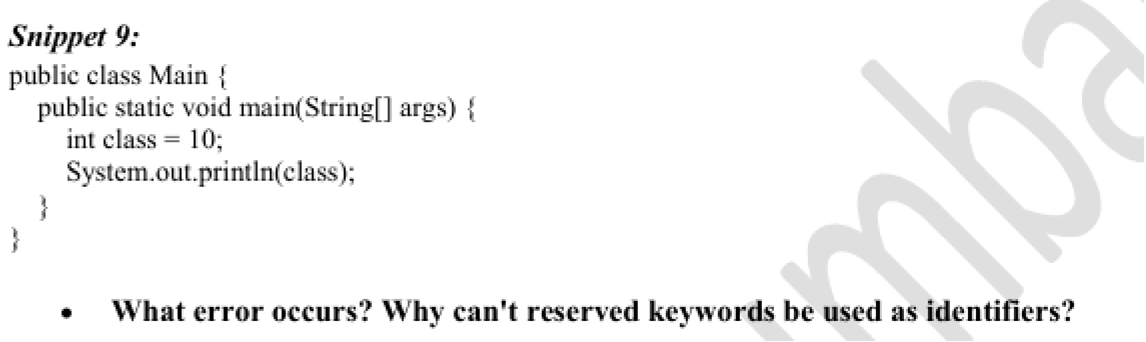
Snippet 8:

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

- **What syntax errors are present? How do they affect compilation?**

Ans

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



Snippet 9:

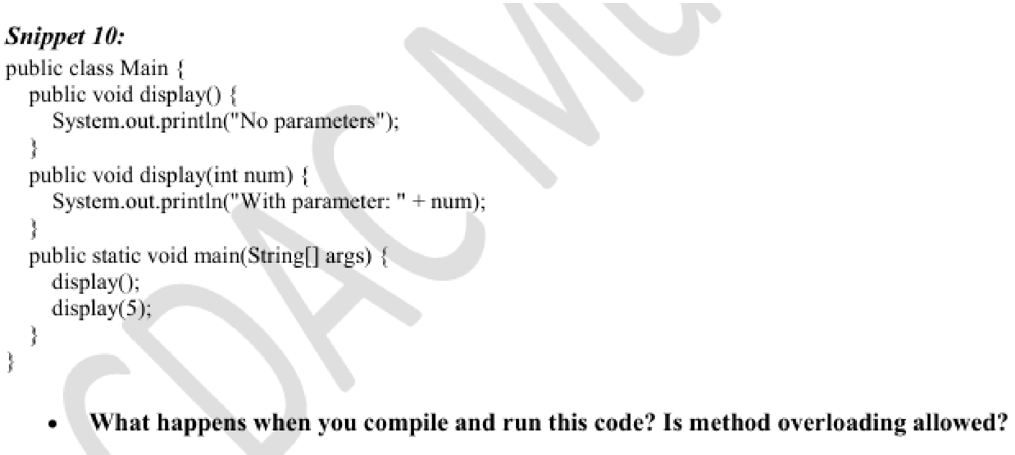
```
public class Main {
    public static void main(String[] args) {
        int class = 10;
        System.out.println(class);
    }
}
```

- **What error occurs? Why can't reserved keywords be used as identifiers?**

Ans

```
ERROR!
/tmp/Rc3MwtYD2R/Main.java:3: error: not a statement
        int class = 10;
        ^
ERROR!
/tmp/Rc3MwtYD2R/Main.java:3: error: ';' expected
```

```
        int class = 10;
             ^
ERROR!
/tmp/Rc3MwtYD2R/Main.java:3: error: <identifier> expected
        int class = 10;
                   ^
ERROR!
/tmp/Rc3MwtYD2R/Main.java:4: error: <identifier> expected
        System.out.println(class);
                             ^
ERROR!
/tmp/Rc3MwtYD2R/Main.java:4: error: illegal start of type
        System.out.println(class);
                              ^
ERROR!
/tmp/Rc3MwtYD2R/Main.java:4: error: <identifier> expected
        System.out.println(class);
                                   ^
ERROR!
/tmp/Rc3MwtYD2R/Main.java:6: error: reached end of file while parsing
}
 ^
7 errors
```



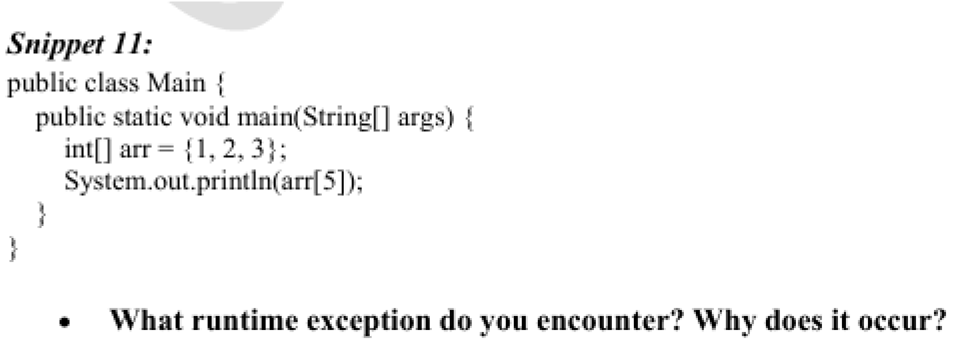
Snippet 10:

```
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);
   }
}
```

- **What happens when you compile and run this code? Is method overloading allowed?**

Ans

```
error: non-static method display() cannot be referenced from a static context

        display();

        ^

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

        display(5);

        ^

2 errors
```



Snippet 11:

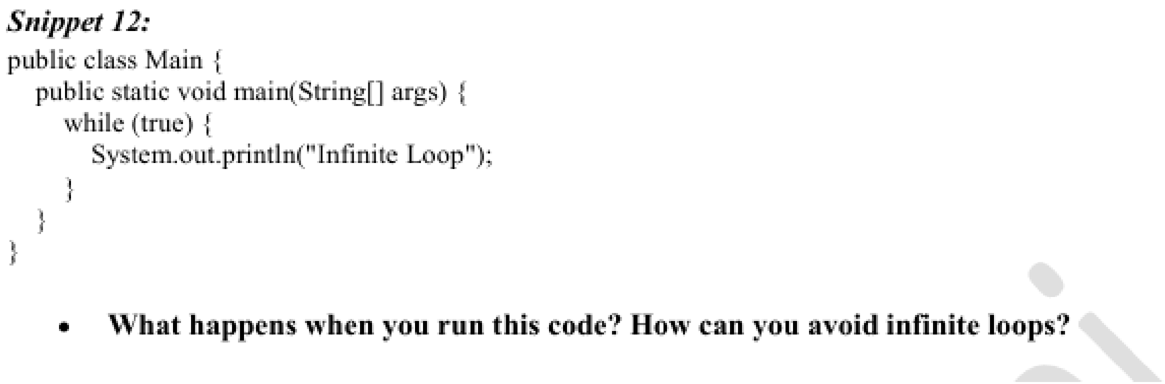
```
public class Main {
   public static void main(String[] args) {
      int[] arr = {1, 2, 3};
      System.out.println(arr[5]);
   }
}
```

- **What runtime exception do you encounter? Why does it occur?**

Ans

ERROR!

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

            at Main.main(Main.java:4)
```



Snippet 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?**

Ans

Infinite Loop

Infinite Loop

Infinite Loop

Infinite Loop

Infinite Loop

Infinite Loop

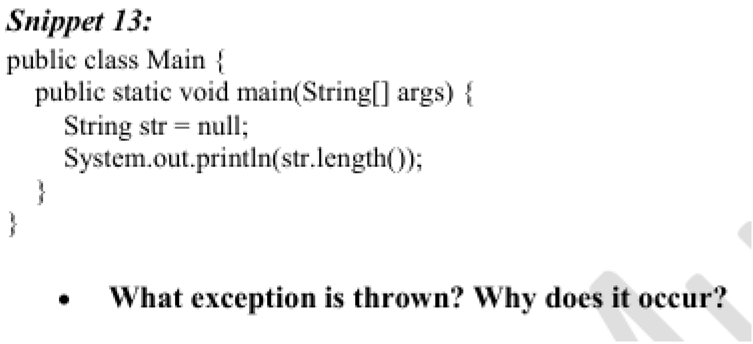
Infinite Loop

Infinite Loop

Infinite Loop

Infinite Loop

Infinite Loop /c



Snippet 13:

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

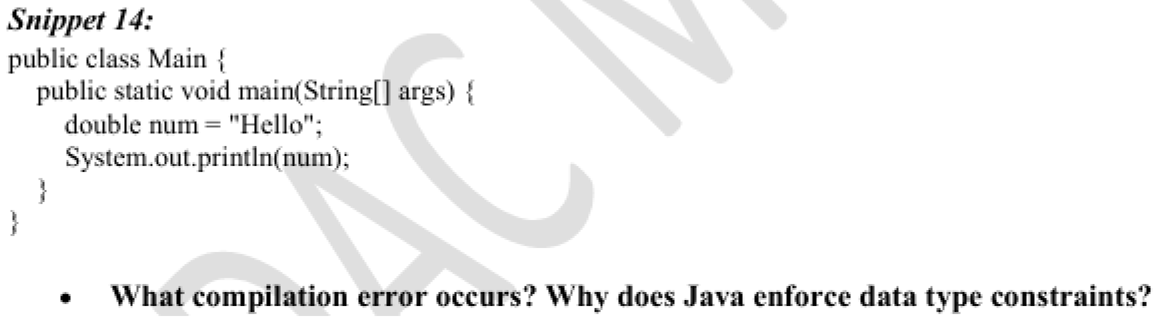
- **What exception is thrown? Why does it occur?**

Ans

ERROR!

Exception in thread "main" java.lang.NullPointerException: Cannot invoke "String.length()" because "<local1>" is null

```
        at Main.main(Main.java:4)
```



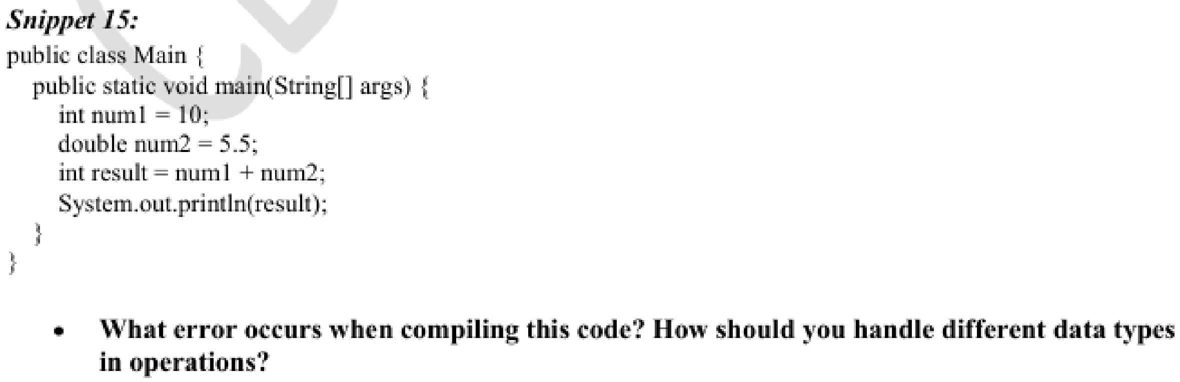
Snippet 14:

```
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?**

Ans

```
error: incompatible types: String cannot be converted to double
    double num = "Hello";
                 ^
1 error
```



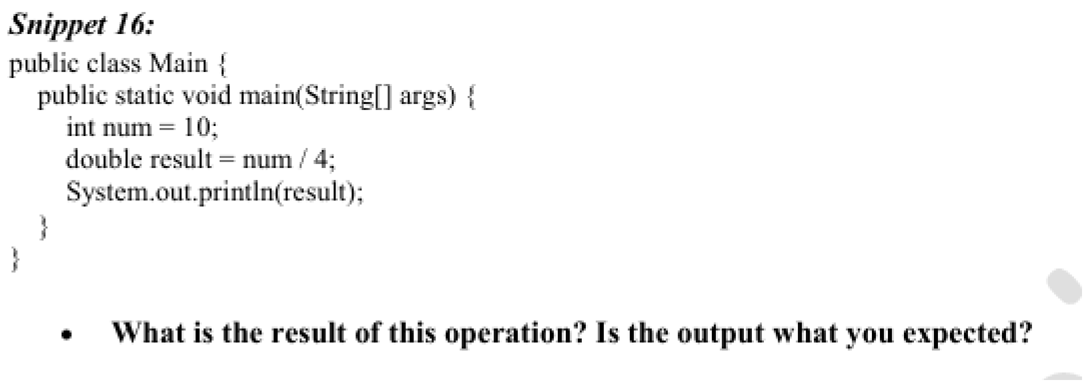
Snippet 15:

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

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

Ans

```
error: incompatible types: possible lossy conversion from double to int
    int result = num1 + num2;
                      ^
1 error
```



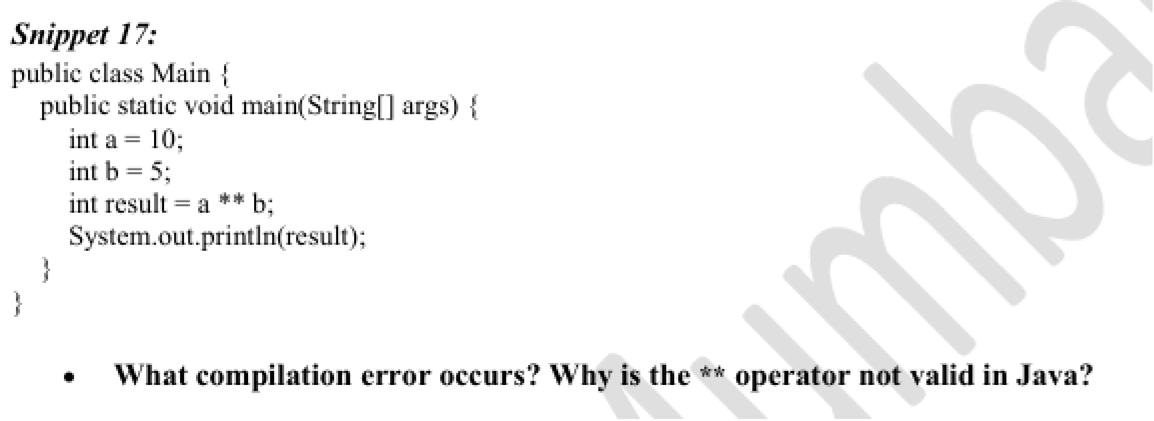
Snippet 16:

```
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?**

Ans

2.0



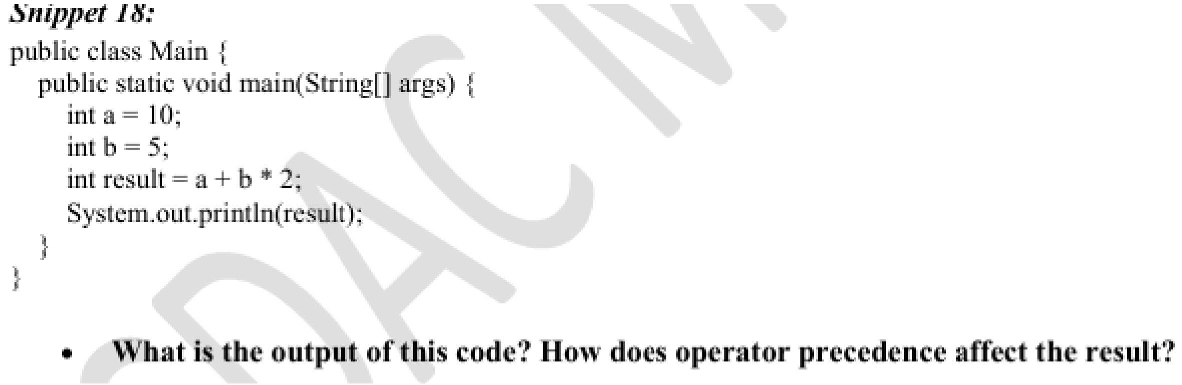
Snippet 17:

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

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

Ans

```
error: illegal start of expression

    int result = a ** b;

                    ^

1 error
```

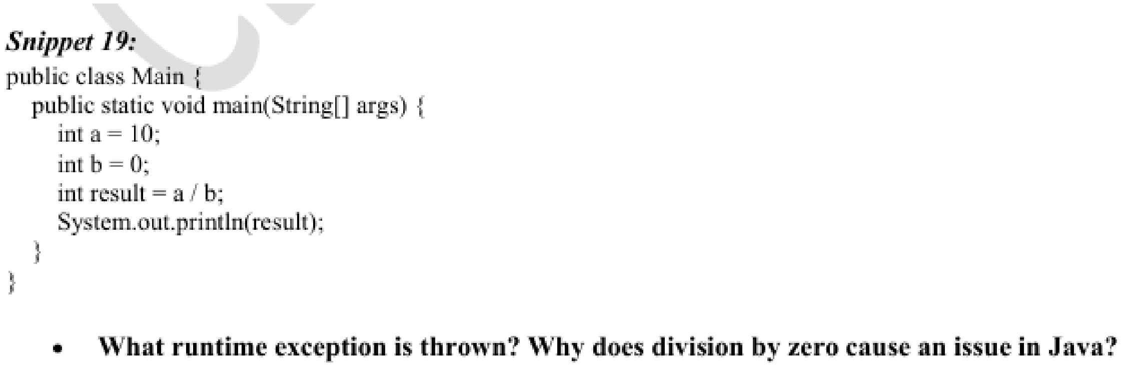


Snippet 18:

```
public class Main {
   public static void main(String[] args) {
      int a = 10;
      int b = 5;
      int result = a + b * 2;
      System.out.println(result);
   }
}
```

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

Ans 20



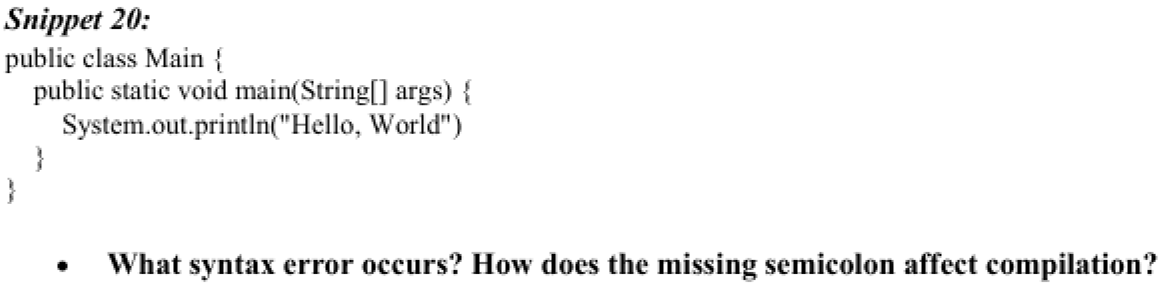
Snippet 19:

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

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

Ans

```
ERROR!
Exception in thread "main" java.lang.ArithmeticException: / by zero
	at Main.main(Main.java:5)
```



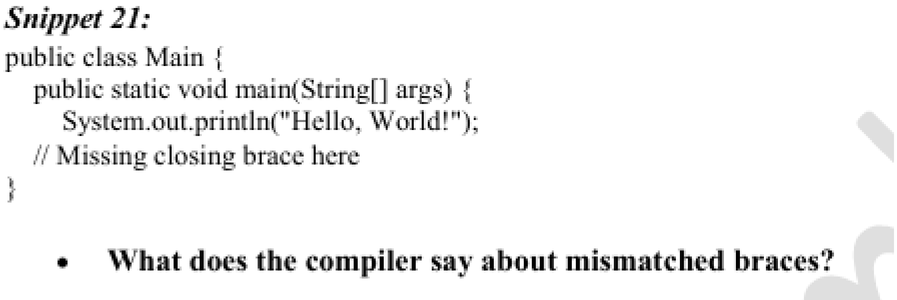
Snippet 20:

```
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?**

Ans

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



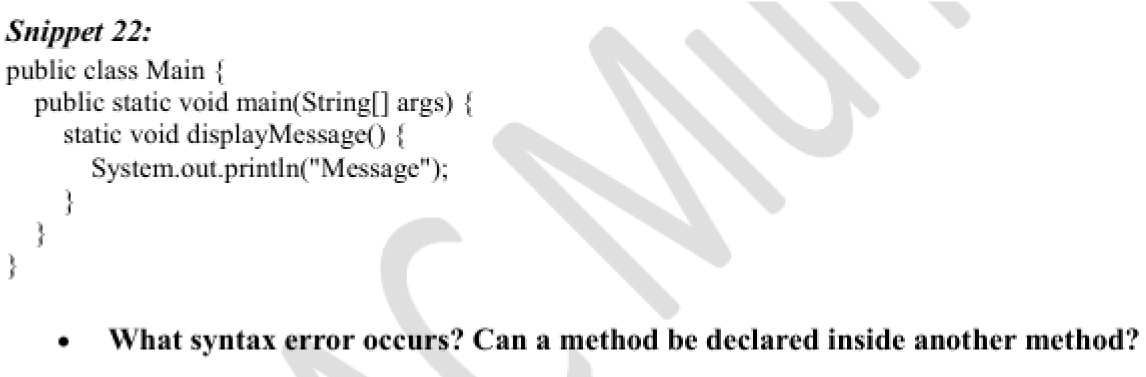
Snippet 21:

```
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?**

Ans

```
error: reached end of file while parsing

}

 ^

1 error
```



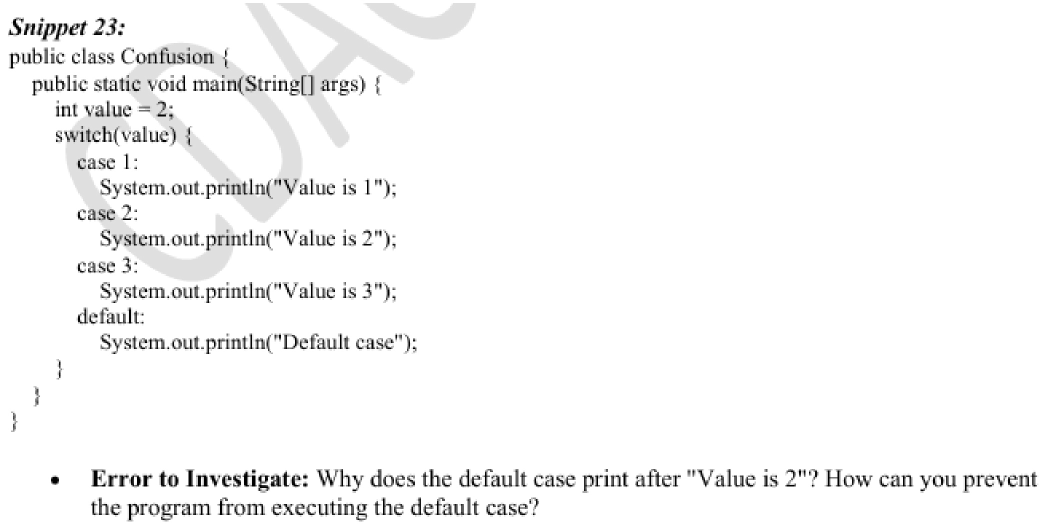
Snippet 22:

```
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?**

Ans:-

```
error: illegal start of expression

        static void displayMessage() {

        ^

ERROR!

error: class, interface, or enum expected

}

^

2 errors
```



Snippet 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");
        }
    }
}
```

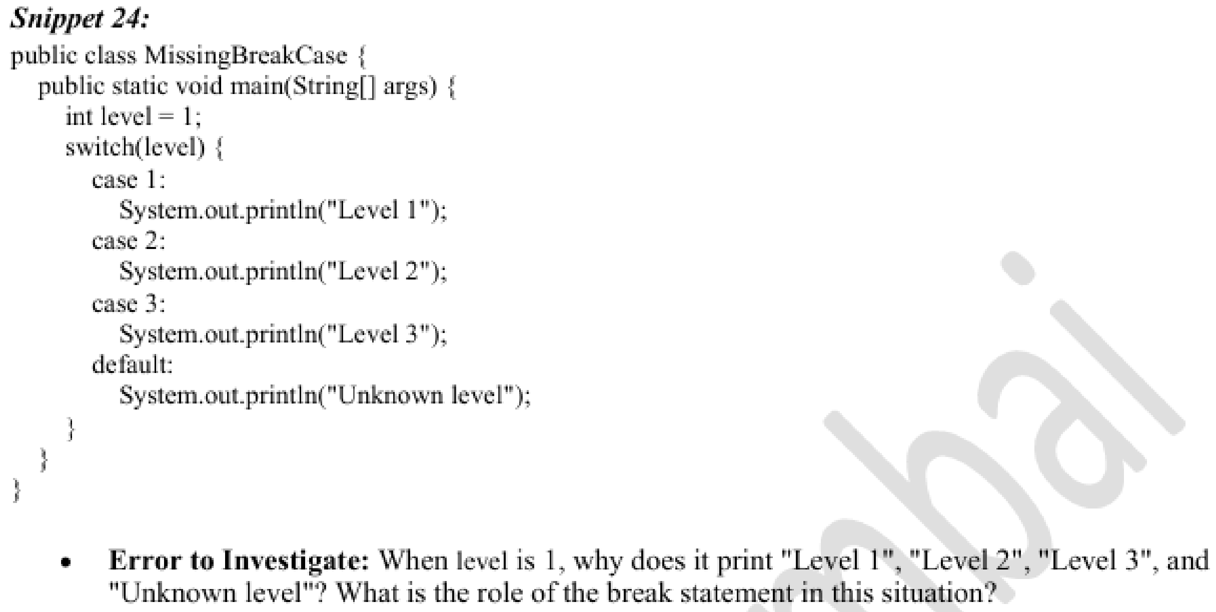
- **Error to Investigate:** Why does the default case print after "Value is 2"? How can you prevent the program from executing the default case?

Ans

Value is 2

Value is 3

Default case



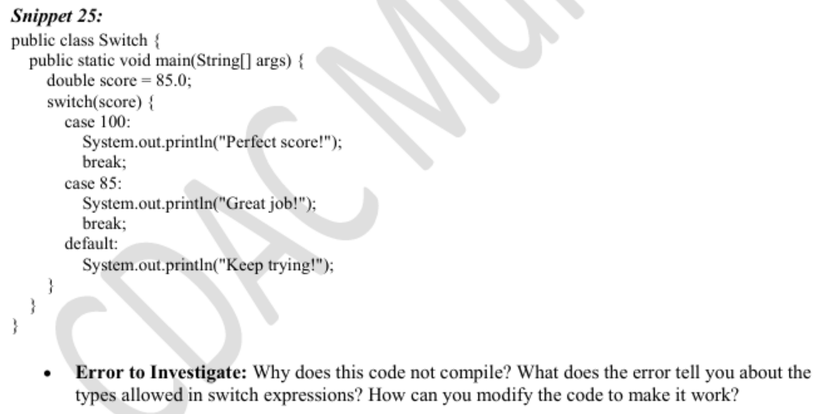
Snippet 24:

```
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?

Ans

```
error: reached end of file while parsing
  }
   ^
1 error
```



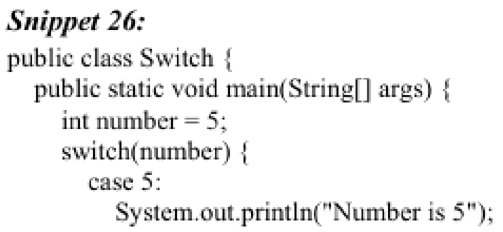
Snippet 25:

```
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?

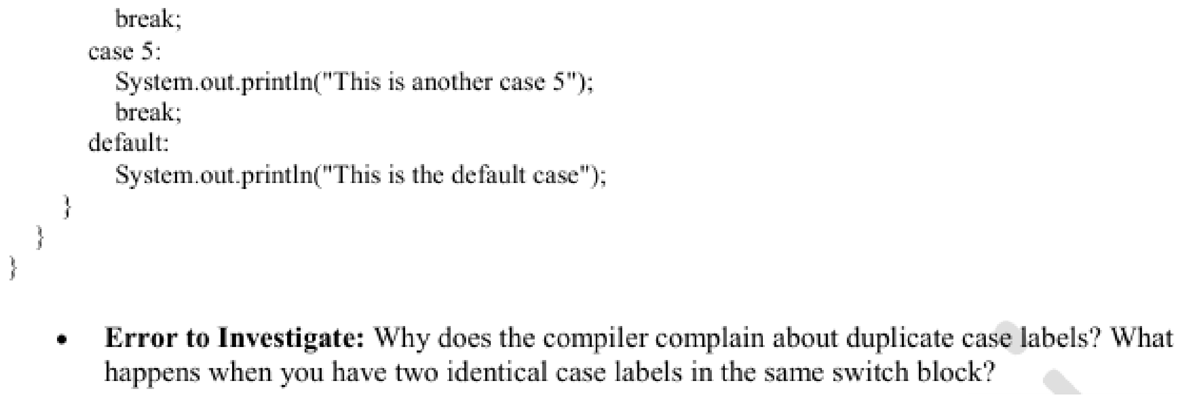
Ans

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

        switch(score) {

              ^

1 error
```



Snippet 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?

Ans

```
error: duplicate case label

            case 5:

            ^

1 error
```