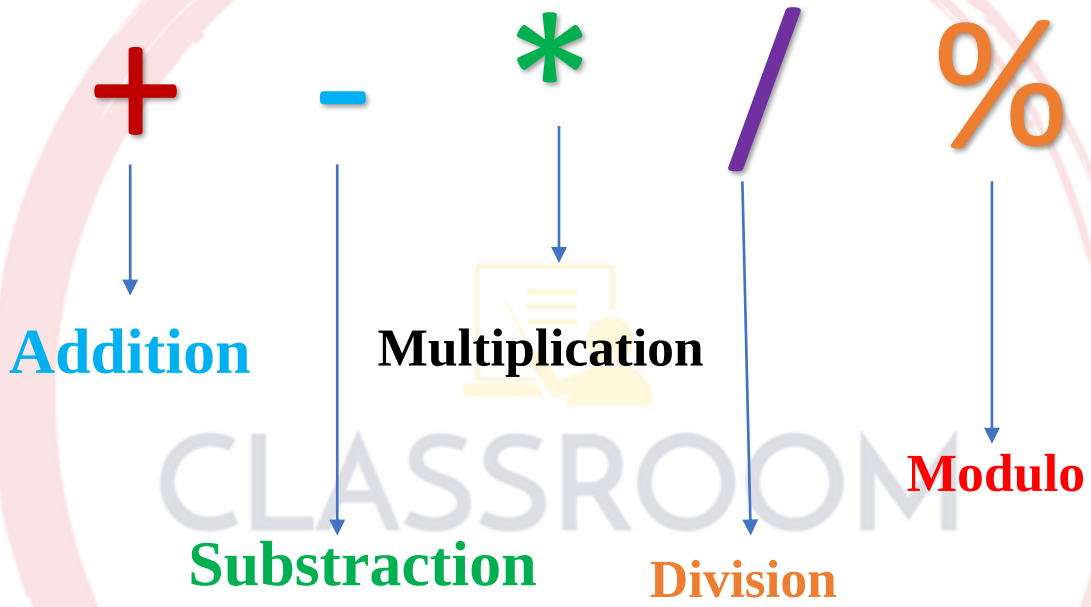
# Operators & Assignment

1. Increment & Decrement operators
2. Arithmetic operators
3. String concatenation operators
4. Relational Operators
5. Equality Operators
6. instanceof Operators
7. Bitwise Operators
8. Short circuit Operators
9. Type cast Operators
10. assignment Operators
11. Conditional Operators
12. new Operators
13. [ ] Operators
14. Precedence of java Operators
15. Evaluation order of java Operands
16. new Vs newInstance()
17. Instanceof Vs isInstance()
18. ClassNotFoundException Vs NoClassDefFoundError

# Arithmetic Operators




```
type of result = max(int, type of a, type of b)
```

# Arithmetic Operators

```
byte  x = 10;

byte y = 2;
```

max(int, type of a, type of b)

x + y  max(int, byte, byte ) → Datatype of <result> integer

```
int result = x + y;
```

```
byte result = x + y;
```
X

**Example:**

```
int x = 9;
long y = 6L;
int  z = x + y ;
```
X // line 3rd compilation error

max( int, int ,long )

```
long z = x + y;   // o/p 15
```

# Arithmetic Operators

Expression:

result = a + b;

max(int, type of a, type of b)

| Type of a | Type of b | Type of result |
|---|---|---|
| byte | byte | int |
| byte | short | int |
| byte | int | int |
| char | char | int |
| Char | int | int |
| byte | char | int |
| int | long | long |
| float | double | double |
| long | long | long |
| long | float | float |
| int | float | float |

# Arithmetic Operators

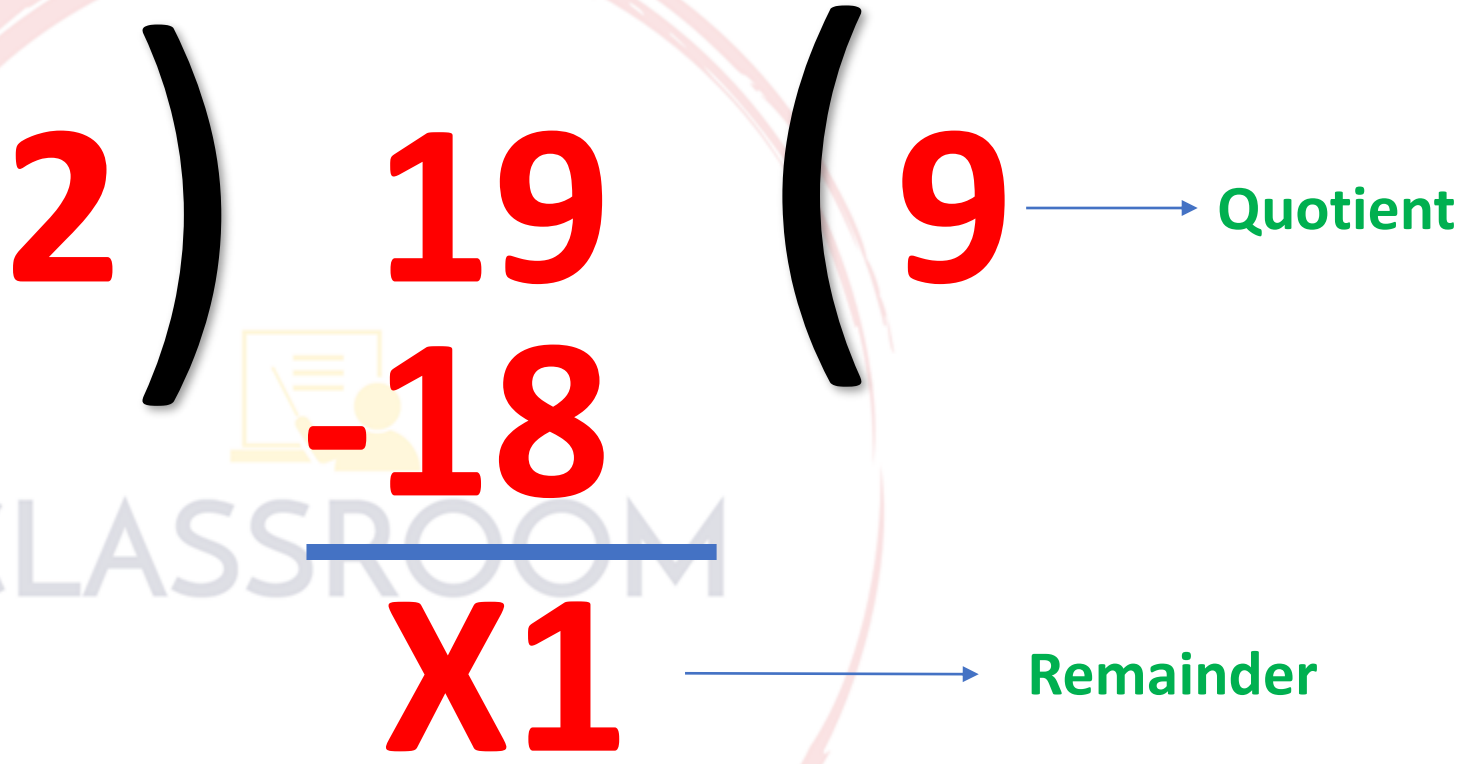
```
int x = 19 ;
int y = 2 ;
```

**Case : 1**

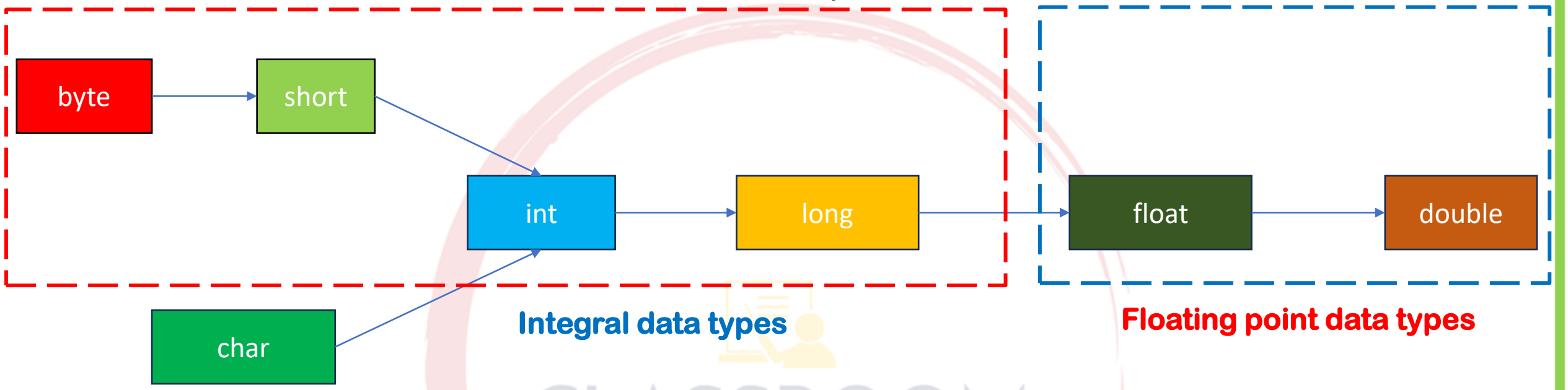
```
int  z = x / y ;  // z = 9
```

**Case : 2**

```
int  z = x  %  y ; // z = 1
```



# Arithmetic Operators






**Integral data types :**

**1. There is no way to represents infinity as a result in Integral data types.**

Example:

```
System.out.println(19/0);
```
RE : ArithmeticException / by zero

**2. There is no way to represent undefine the result in Integral data types.**

Example:

```
System.out.println(0/0);
```
RE : ArithmeticException / by zero

# Arithmetic Operators

## Floating point data types :

For the Float and Double classes contains following constants:

1. POSITIVE_INFINITY
2. Negative_INFYNITY
3. NaN

### 1. We represents infinity as a result in floatingpoint data types.

Example:

```
System.out.println(19 / 0.0 );
```

**Output : Infinity**

### 2. We can also represent undefine as a result in Floating point data types.

Example:

```
System.out.println(0 / 0.0 );
```

**Output : NaN**

# Arithmetic Operators