

Logic Building Session Day 2: March 2022

Kiran Waghmare

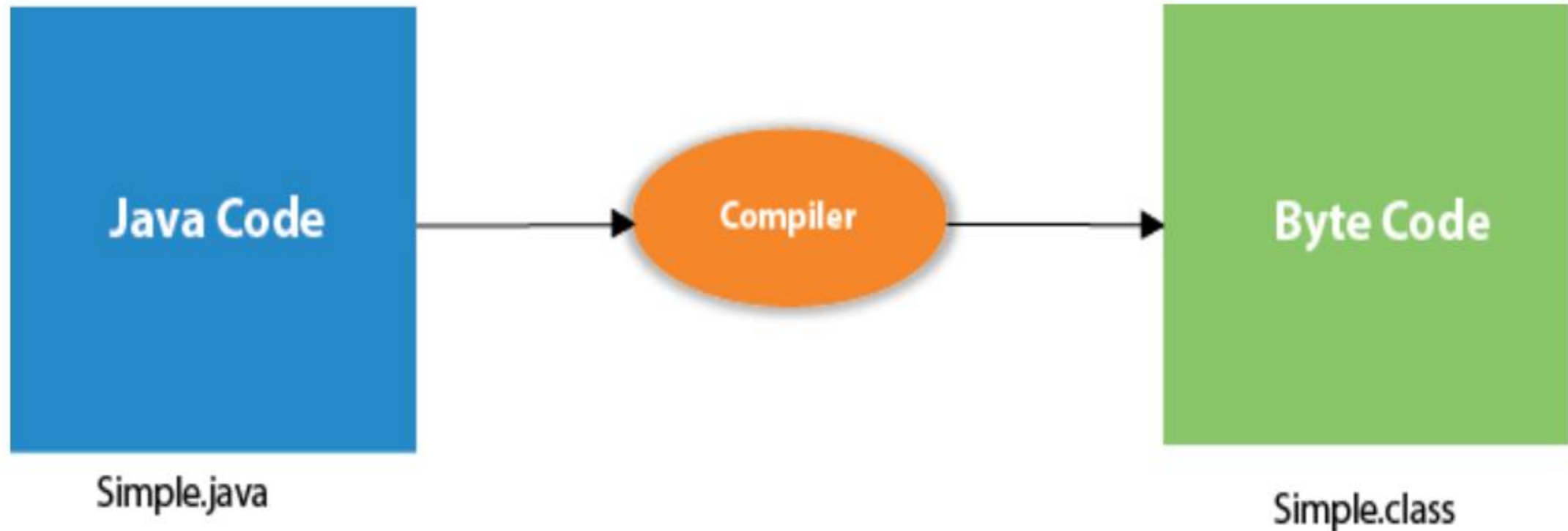


Program 1:

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

Compilation Flow:

When we compile Java program using javac tool, java compiler converts the source code into byte code.



Exercise:

Identify the valid java main method signature

1. `public static void main(String[] args)`
2. `public static void main(String []args)`
3. `public static void main(String args[])`
4. `public static void main(String... args)`
5. `static public void main(String[] args)`
6. `public static final void main(String[] args)`
7. `final public static void main(String[] args)`
8. `public void main(String[] args)`
9. `static void main(String[] args)`
10. `public void static main(String[] args)`
11. `abstract public static void main(String[] args)`

Java Tokens

- **Tokens - The smallest individual unit of program are known as Tokens.**
- **Java Program – It is a collection of Tokens , comments and white spaces. It contains 5 types of tokens:**

Reserved words – keywords

- 50 keywords
- Having specific meaning – we cannot use them as names for variables ,class name etc
- Always lower case letters, case sensitive
- E.g., abstract, case, short, super etc

Identifiers – a

- Programmer designed tokens
- Used for naming classes, methods, variables, labels, packages, interfaces in a program

- **Rules-**
 1. Have alphabets, digits and _ and \$
 2. Not begin with digit
 3. Uppercase & lowercase letters are distinct
 4. Can be of any length

- **Literals –**

- Sequence of character
- Represents constant value to be stored in variable
- 5 – types- Integer, Floating-point, Character, String and Boolean

- **Operators –**

- Symbol that takes one / more arguments & operates on them to produce a result.

- **Separators –**

- Group of code are divided & arranged
- i.e., (), { }, [], ; , , & .

Java Tokens

- **Tokens** - The smallest individual unit of program are known as Tokens.
- **Java Program** - It is a collection of Tokens , comments and white spaces. It contains 5 types of tokens:

Reserved words – keywords

- 50 keywords
- Having **specific meaning** – we cannot use them as names for variables ,class name etc
- Always **lower case letters**, case sensitive
- E.g., abstract, case, short, super etc

Identifiers – a

- Programmer designed tokens
- Used for naming classes, methods, variables, labels, packages, interfaces in a program

Rules-

1. Have alphabets, digits and _ and \$
2. Not begin with digit
3. Uppercase & lowercase letters are distinct
4. Can be of any length

Literals –

- Sequence of character
- Represents constant value to be stored in variable
- 5 – types- Integer, Floating-point, Character, String and Boolean

Operators –

- Symbol that takes one / more arguments & operates on them to produce a result.

Separators –

- Group of code are divided & arranged
- i.e., (), { }, [], ; , , & .

int:data type
Integer:Class
integer:Identifier

(), [], { }, ; , , & .

Identifiers

n1
n2
+
sum
>)
sum
SUM
Sum
suM


```
class Variable
```

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

```
        int n1, n2, sum; //Declaration of variable
```

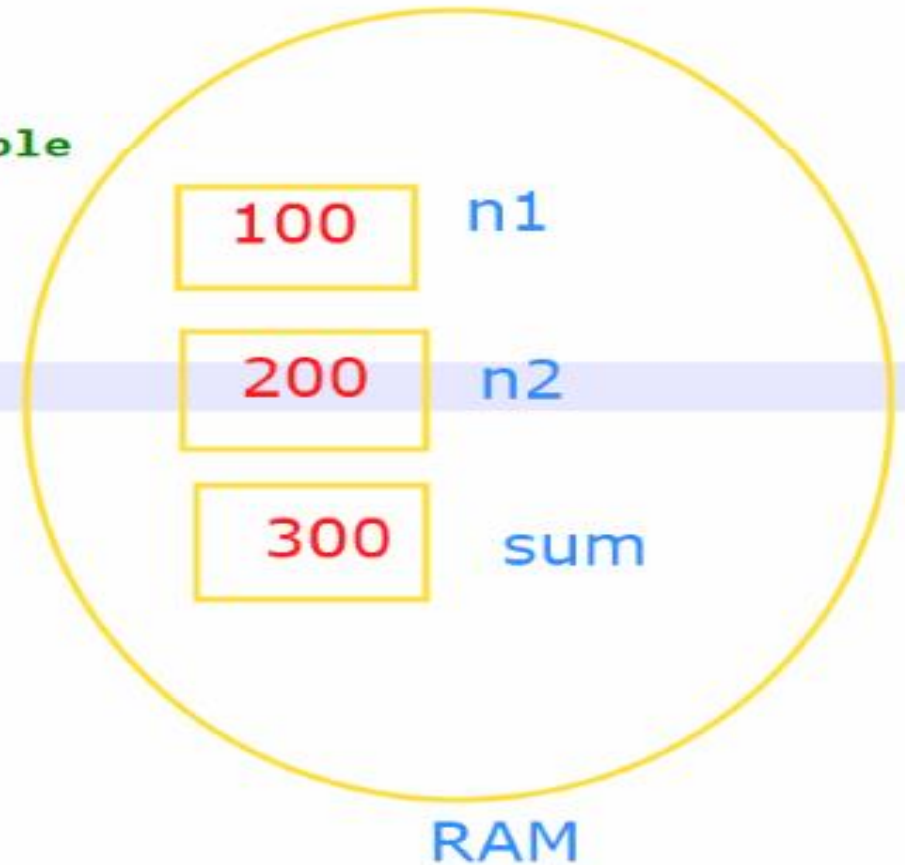
```
        n1=100; //Initialization of variable
```

```
        n2=200;
```

```
        sum=n1+n2;
```

```
        n1+n2=sum; //wrong way to write
```

```
    }  
}
```



class variable

```
{  
    public static void main(String args [])  
    {  
        int n1, n2, sum; //Declaration of variable  
        n1=100; //Initialization of variable  
        n2=200;  
        sum=n1+n2;  
        System.out.println(sum);  
    }  
}
```

100

n1

200

n2

300

sum

RAM

class variable1

```
{  
    public static void main(String args [])  
    {  
        int i = 100;  
        int j = 200;  
  
        int k = i+j;  
  
        System.out.println(k);  
        System.out.println("Sum is = "+k);  
    }  
}
```



Who can see what you share here? Recording



```
class variable1
```

```
{
```

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

```
    {
```

```
        int i = 100;
```

```
        int j = 200;
```

```
        int k = i+j;
```

```
        System.out.println(k);
```

```
        System.out.println("Sum is = "+k);
```

```
        System.out.println("Sum of "+i+" and "+j+"="+k);
```

```
        System.out.println("CDAC Mumbai "+ i+ "% placement for March 2022");
```

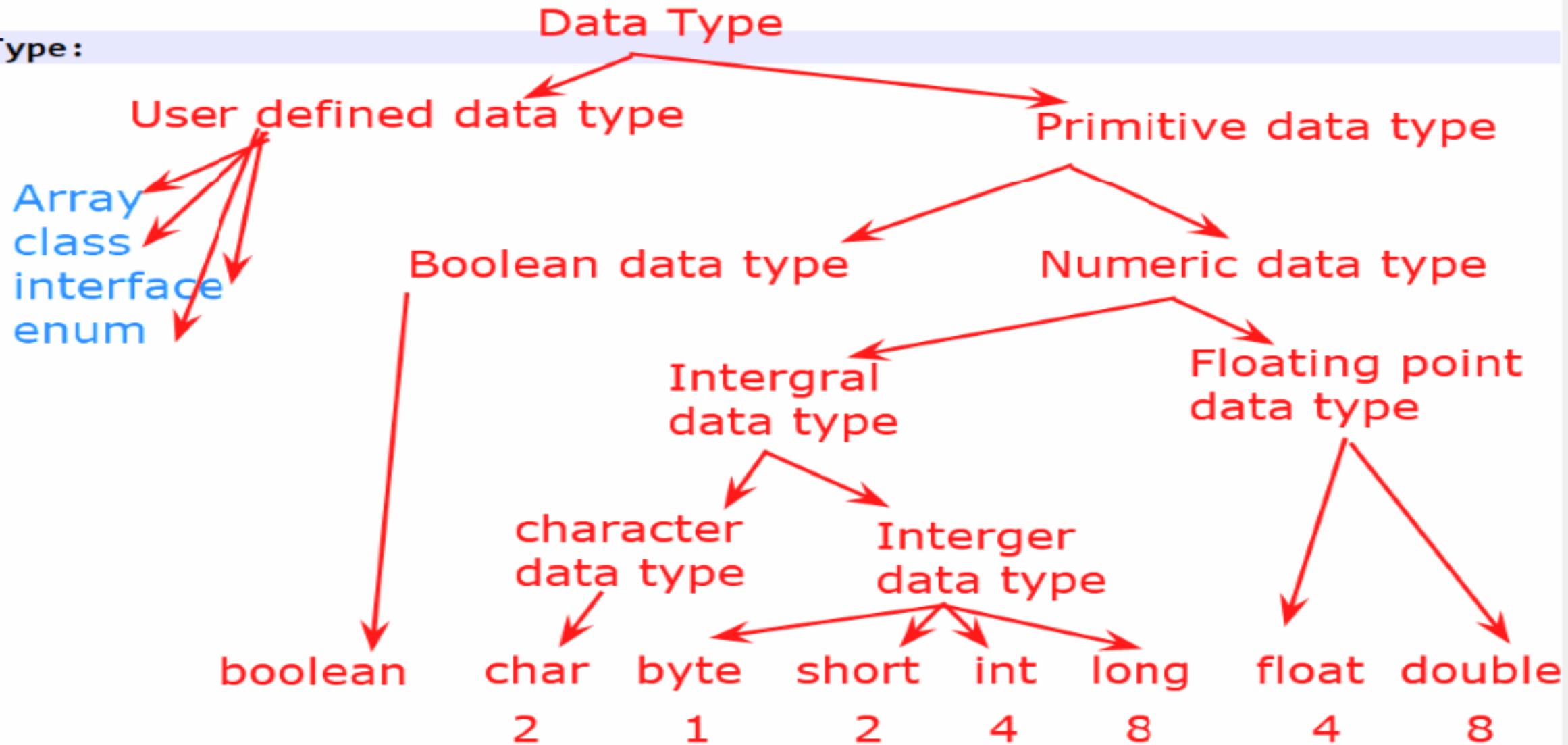
```
    }
```

```
}
```

What is Data Types in Java?

- Data Types in Java are defined as specifiers that allocate different sizes and types of values that can be stored in the variable or an identifier.
- Java has a rich set of data types. Data types in Java can be divided into two parts :
 1. Primitive Data Types :- which include integer, character, boolean, and float
 2. Non-primitive Data Types :- which include classes, arrays and interfaces.

Data Type:



Unicode system

min : \u0000

max : \uFFFF

```
class Primitive
{
    public static void main(String args [])
    {
        boolean b = true; //true/false
        byte by =127;
        short sh =555;
        int i =99999;
        long l = 9999999999L;
        System.out.println(b);
        System.out.println(by);
        System.out.println(sh);
        System.out.println(i);
        System.out.println(l);
    }
}
```

Command Prompt

```
C:\CDAC22>java Primitive
true
127
555
99999
9999999999
C:\CDAC22>
```

```
public static void main(String args [])  
{    //upcasting  
    int i =100;  
    float f = i;  
    System.out.println(i);  
    System.out.println(f);  
  
    float i1 =100.734895894984598F;  
    double f1 = i1;  
    System.out.println(i1);  
    System.out.println(f1);  
    //downcasting  
  
    double i2 =100.734895894984598;  
    float f2 =(float)i2;  
    System.out.println(i2);  
    System.out.println(f2);  
  
    double i3 =100.7;  
    float f3 = (float)i3;  
    System.out.println(i3);  
    System.out.println(f3);  
}
```

