Fundamentals of Object Oriented Programming

CSN- 103

Dr. R. Balasubramanian

Associate Professor

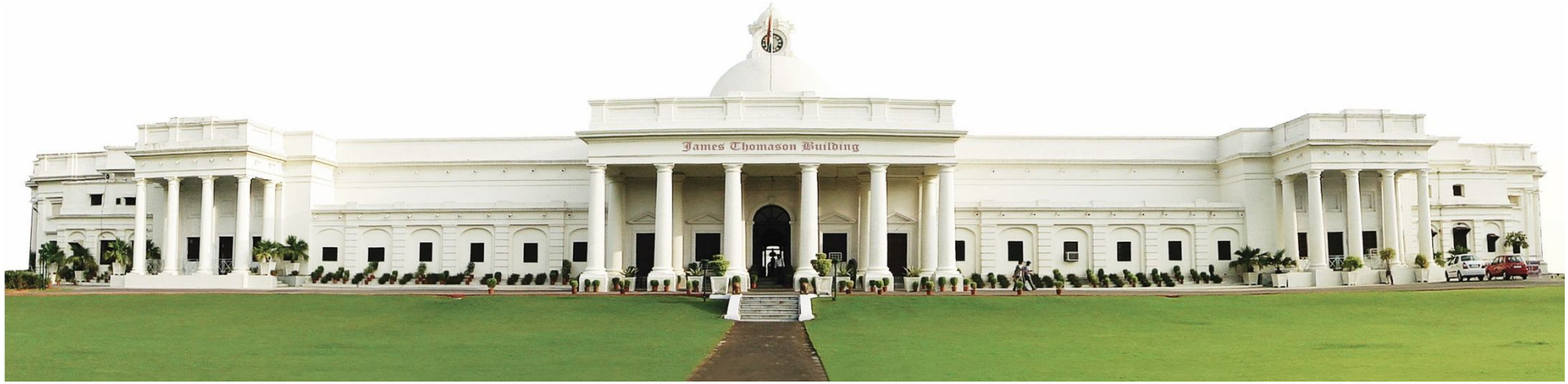
Department of Computer Science and Engineering

Indian Institute of Technology Roorkee

Roorkee 247 667

balarfcs@iitr.ac.in

https://sites.google.com/site/balaiitr/



Floating Point Types

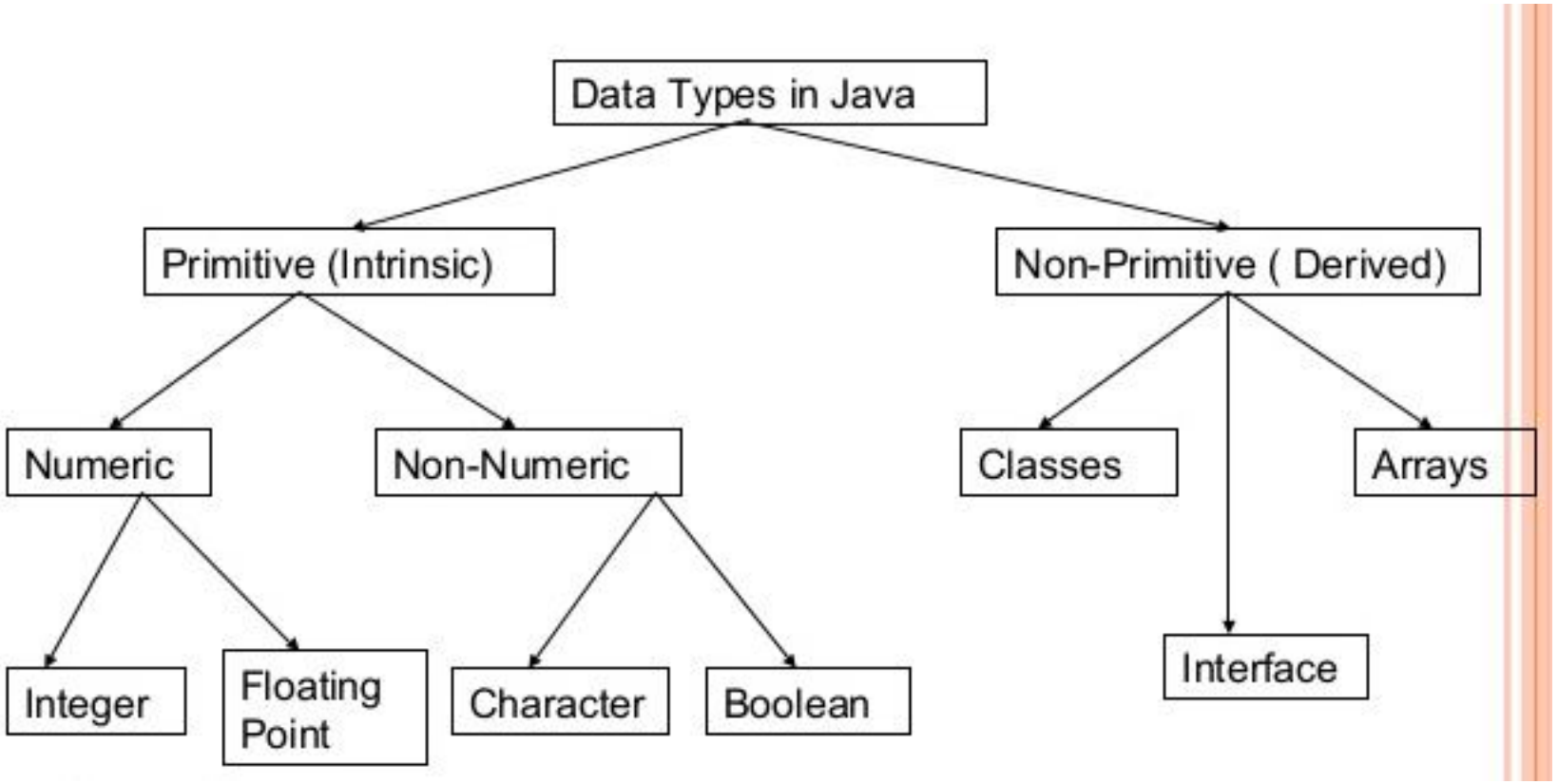
- Integer types can hold whole numbers only.
- We use floating point type to hold the numbers having fractional parts such as 27.59 and -1.342.
- There are two types of floating point storage in Java.
- Floating point numbers are treated as double-precision quantities. To force them to be in single precision mode, we must append f or F to the numbers.

1.23f *7.56923e5F*

Type	Size
float	4 bytes
double	8 bytes

- *Not-a-Number (NaN): Divide by zero and operand is NaN*

Data Types


Data Types in Java
Primitive (Intrinsic)
Non-Primitive (Derived)
Numeric
Non-Numeric
Classes
Arrays
Integer
Floating Point
Character
Boolean
Interface

Declaration of variables

- Declaration of a variable does three things:
  - It tells the compiler what the variable name is.
  - It specifies what type of data the variable will hold.
  - The place of declaration (in the program) decides the scope of the variable.
- The general form of variable declaration is:

  type variable1, variable2,……,variableN;

  Like

```
int      count;
float    x, y;
double   pi;
byte     b;
char     c1, c2, c3;
```

Giving values to variables

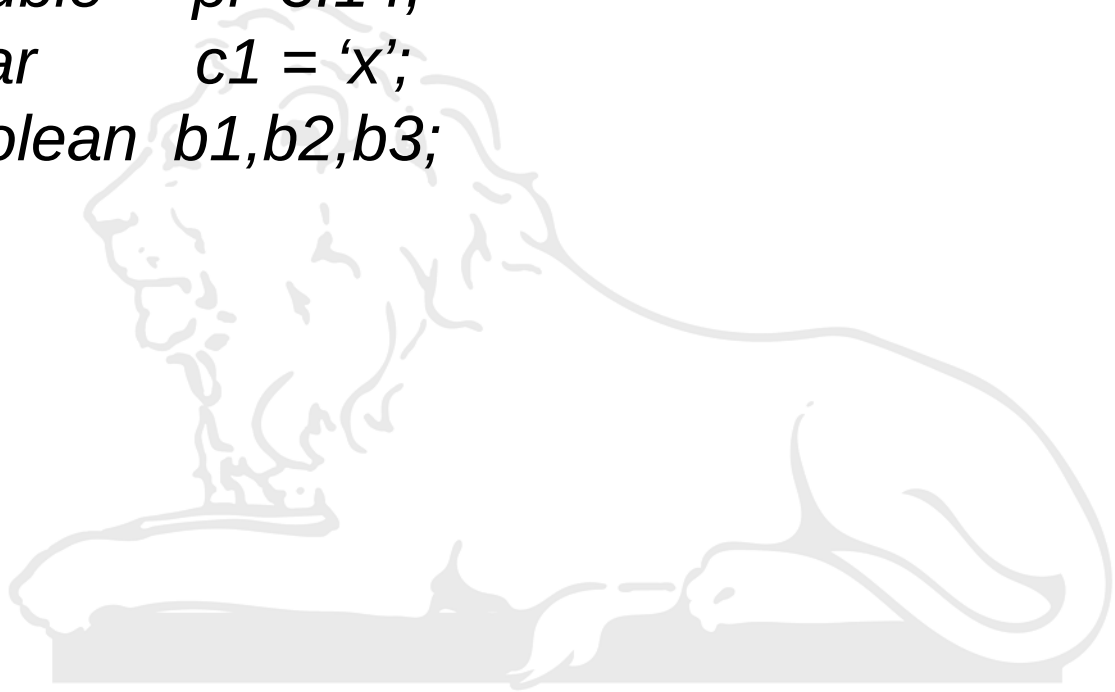
- This can be achieved in two ways:
  - By using an assignment operator
  - By using a read statement (getting input from the keyboard)
- **Assignment statement:**

```
variableName = value;
```

for example:

```
initialValue = 0;
finalValue =  100;
yes        =  'x';
```

```
int       count  = 10;
float     x =5.2f, y=2.5f;
double    pi=3.14;
char      c1 = 'x';
boolean   b1,b2,b3;
```



```java
import java.util.*;
public class Bool{
    public static void main(String []args){
        boolean b1,b2,b3;

    Scanner n = new Scanner(System.in);
        b1 = n.nextBoolean();
        //b1 = true;  // Assigning Value
        b2 = false; // Assigning Value
        b3 = b2;    // Assigning Variable

    System.out.println(b1); // Printing Value
    System.out.println(b2); // Printing Value
    System.out.println(b3); // Printing Value
    }
}
```

```
sh-4.4$ javac Bool.java
sh-4.4$ java Bool
false
false
false
false
sh-4.4$
```

https://ideone.com/Lg3Lcd

Scope of Variables

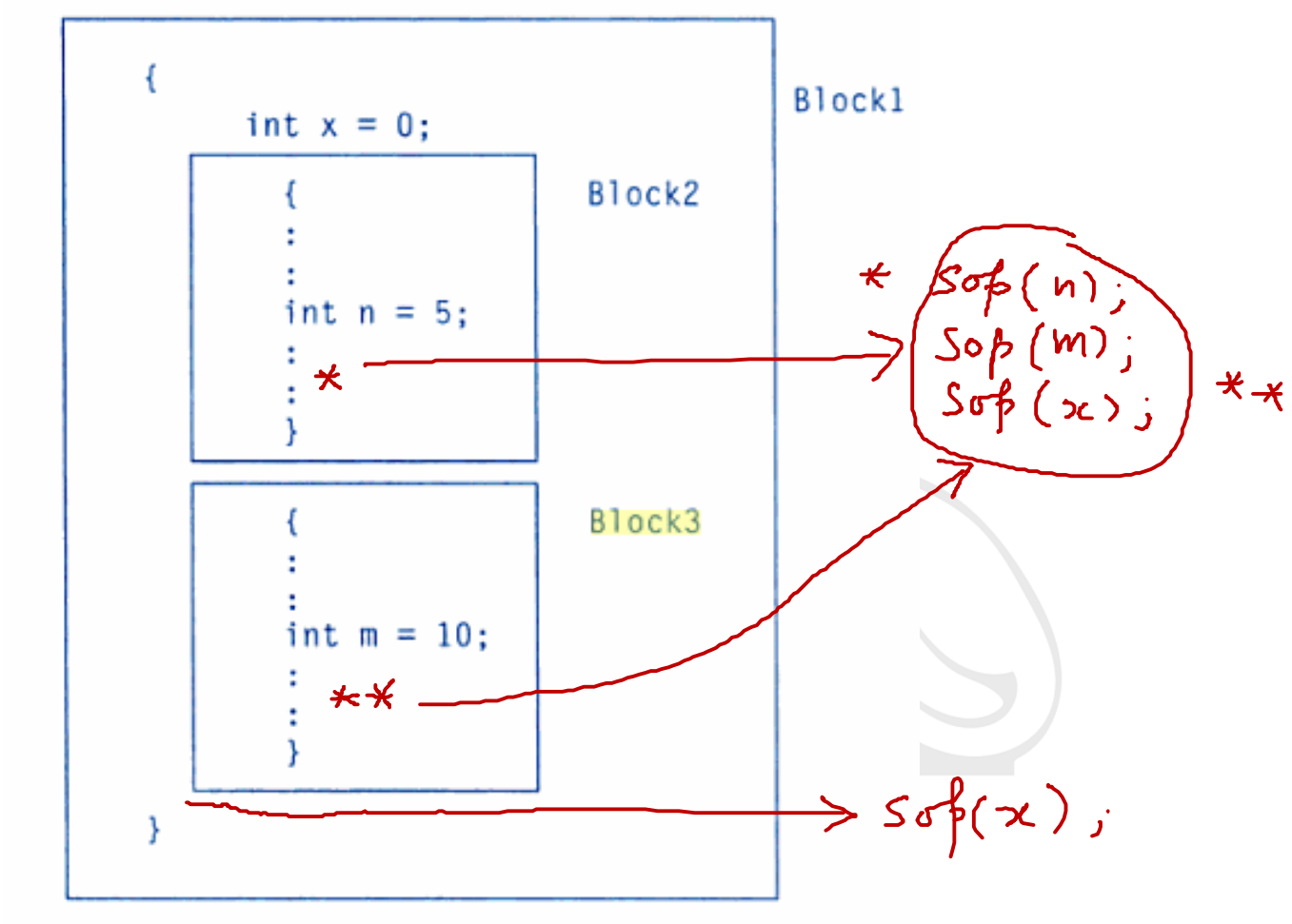
- **Instance variables**
- **Class variables**
- **Local variables**

Instance and class variables declare inside a class. In particular, instance variables are created when the objects are instantiated and therefore they are associated with the object.

Class variables are global to a class and belong to the entire set of objects that class created.

Variables declared and used inside methods are called local variables. They are called so because they are not available for use outside the method definition.

Nested Program Blocks



Symbolic Constants

```
final   int     STRENGTH=100;
final   int     PASS_MARK=50;
final   float   PI=3.14159;
```

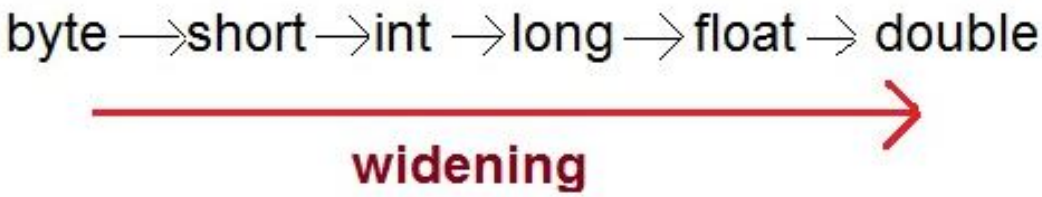
const int a = 100;

Type Casting

```
int x = 10;
byte y = (byte) x;
```

```
byte b = 4;
double e;
e = b;
sop(e);
```

- Widening Casting(Implicit)



- Narrowing Casting(Explicitly done)

Type Casting

```
1  public class Typec1{
2
3      public static void main(String []args){
4         int i = 100;
5       long l = i;   //no explicit type casting required
6       float f = l;  //no explicit type casting required
7       System.out.println("Int value "+i);
8       System.out.println("Long value "+l);
9       System.out.println("Float value "+f);
10     }
11 }
12
```

Terminal

```
sh-4.3$ javac Typec1.java
sh-4.3$ java Typec1
Int value 100
Long value 100
Float value 100.0
sh-4.3$
```

Type Casting

```
1  public class Typec2
2  {
3      public static void main(String[] args)
4      {
5        double d = 100.04;
6        long l = (long)d;  //explicit type casting required
7        int i = (int)l;   //explicit type casting required
8
9        System.out.println("Double value "+d);
10       System.out.println("Long value "+l);
11       System.out.println("Int value "+i);
12
13     }
14
15 }
```

floor [d] → 100

ceil [d] → 101

```
Terminal
sh-4.3$ javac Typec2.java
sh-4.3$ java Typec2
Double value 100.04
Long value 100
Int value 100
sh-4.3$
```

ceil ← [100.04] → 101

floor ← [100.04] → 100

print and println

```java
public class HelloWorld1{

    public static void main(String []args){
        System.out.print("Hello World");
    }
}

```

```
Terminal
sh-4.3$ javac HelloWorld1.java
sh-4.3$ java HelloWorld1
Hello Worldsh-4.3$
```

```java
public class HelloWorld2{

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

```

```
Terminal
sh-4.3$ javac HelloWorld2.java
sh-4.3$ java HelloWorld2
Hello World
sh-4.3$
```

Default Value of Data Types in Java

Data Type	Default Value (for fields)
byte	0
short	0
int	0
long	0L
float	0.0f
double	0.0d
char	‘u0000’
String (or any object)	null
boolean	false

in c++ | int a;
cout << a;

Default Value of Data Types in Java

```
1  public class DefaultValue {
2    static boolean bool;
3    static byte by;
4    static char ch;
5    static double d;
6    static float f;
7    static int i;
8    static long l;
9    static short sh;
10   static String str;
11
12   public static void main(String[] args) {
13     System.out.println("Bool :" + bool);
14     System.out.println("Byte :" + by);
15     System.out.println("Character:" + ch);
16     System.out.println("Double :" + d);
17     System.out.println("Float :" + f);
18     System.out.println("Integer :" + i);
19     System.out.println("Long :" + l);
20     System.out.println("Short :" + sh);
21     System.out.println("String :" + str);
22   }
23 }
```

Terminal

```
sh-4.3$ javac DefaultValue.java
sh-4.3$ java DefaultValue
Bool :false
Byte :0
Character:
Double :0.0
Float :0.0
Integer :0
Long :0
Short :0
String :null
sh-4.3$
```