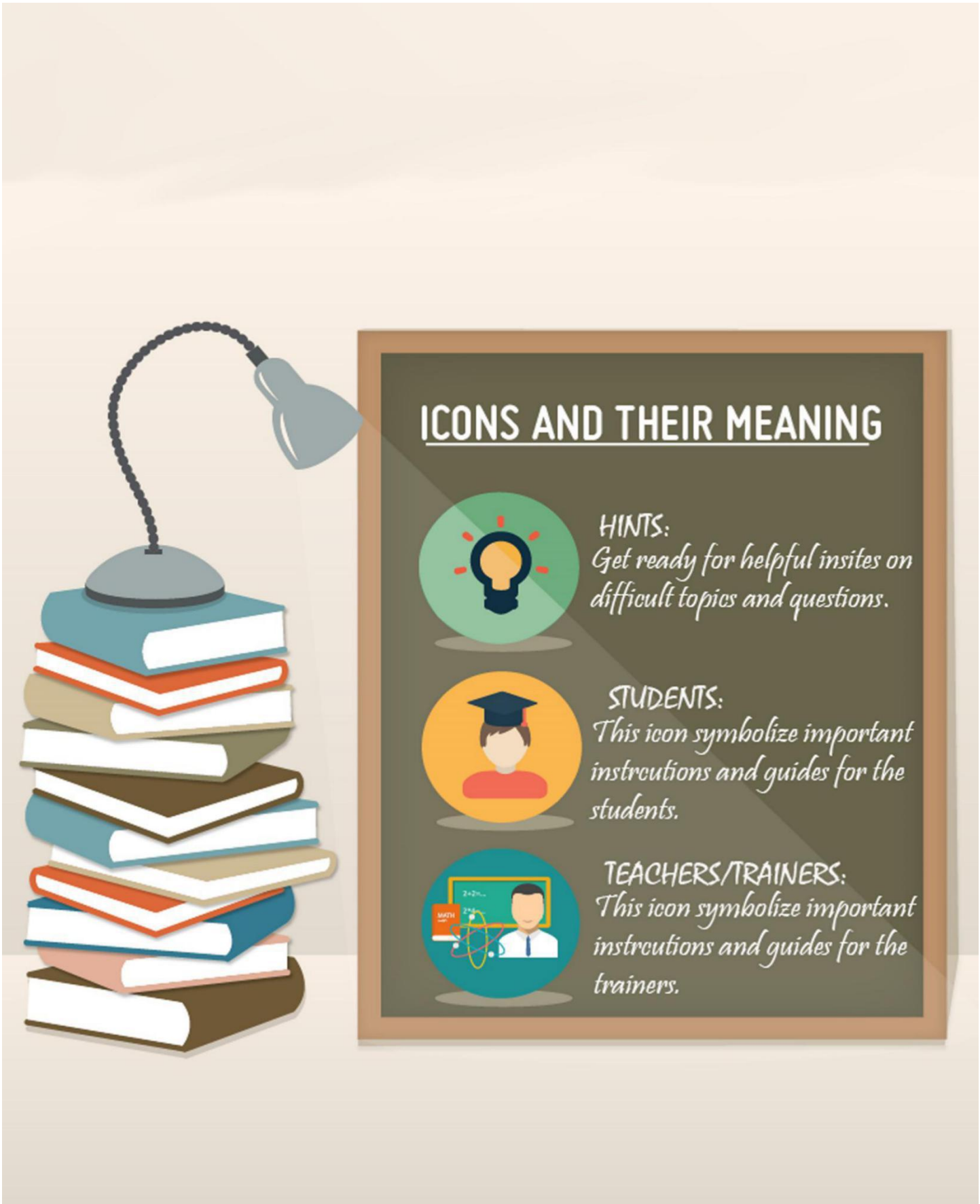
CORE JAVA

MANUAL V8.3

MODULE CODE:

ANUDIP FOUNDATION


ICONS AND THEIR MEANING
HINTS:
Get ready for helpful insites on difficult topics and questions.
STUDENTS:
This icon symbolize important instrcutions and guides for the students.
TEACHERS/TRAINERS:
This icon symbolize important instrcutions and guides for the trainers.

Module 1: Java Fundamental and Programming Concepts

Chapter 5

Objective: After completing this lesson you will be able to : * Get familiar with variables * Gain an idea about Java type casting	**Materials Required:** 1. Computer 2. Internet access
Theory Duration: 120 minutes	**Practical Duration:** 0 minute
Total Duration: 120 minutes	

Chapter 5

5.1 Variable (Local, Static, Global),

Java Variables

A variable in Java refers to a container which holds a value during java program execution. A Java variable has to have a data type assigned to it. It is a memory piece capable of containing a data value. In simpler terms, a variable is the name of a memory location. The value of a variable can be changed.

Example of a Java variable – int data=60;→ 'data' is the variable here and 'int' is the data type.

Variables in java can be classified into three types

i) local variable

ii) instance variable

iii) static variable

i) Local Variable

A local variable is a variable declared within the body of a method. This variable type can be used only within its designated method. Only a variable's own method is aware of it, and not other methods. Using a 'static' keyword to define a local variable is not possible. Default values are not assigned to local variables.

A block {.} is used to store a statement of local variable declaration.

Local variable syntax example –

```
int area()
   {
```

```
int length=20; → local variable
int breadth = 10; → local variable
int rectarea = length*breadth; → local variable
return rectarea;
}
```

ii) Instance Variable

An instance variable is a variable declared within a class, but outside of a method body. It is similar to a class variable in Java. The value of an instance variable is specific to an 'instance', and not shared with other instances. Access modifiers can be assigned to these variables. An instance variable cannot be declared as 'static'. This variable is created when an object is created, and destroyed if the object is destroyed.

Instance variable declaration syntax example –

```
class Taxes
{
int count; → Count is an Instance variable
/*...*/
}
```

iii) Static Variable

A static variable is a variable declared with a 'static' keyword. A static variable's copy can be shared with all instances of a class. Static variable memory allocation only happens once, during the loading of a class in the memory This variable type is not local. It is used for achieving shared properties of all objects.

Static variable syntax example – static <DataType> <variable_name>;

An example with all 3 variables (static, local and instance) –

```
class B{
int data=60;  → instance variable
static int m=120;  → static variable
void method(){
int n=100; → local variable
}
}
```

*** Difference between variable declaration and variable assignment**

Variable declaration	Variable assignment
The declaration of a value to a variable	The assignment of a value to a variable
Declaring a variable stores a value in the memory assigned to a variable.	Assigned variable signals a compiler to assign variable memory in stack.

5.2 Type Casting (Wider & Narrow),

Type casting

Type Casting in Java

Type casting in Java is used for converting a variable or object type to another. It involves the assigning of a primitive data type value to another primitive data type.

Syntax – dataType variableName = (dataType) variableToConvert;

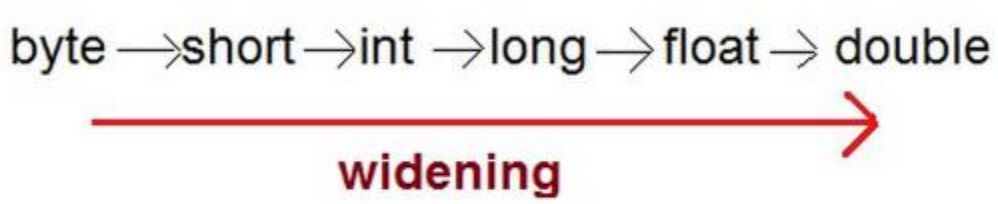
The types of casting based on directions are –

i) **narrowing –** larger to smaller data type casting

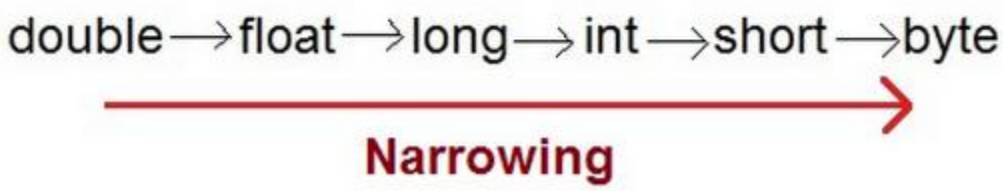
ii) **widening –** smaller to larger data type casting

Diagram illustrating narrowing and widening –

- Widening Casting(Implicit)



- Narrowing Casting(Explicitly done)



i) **Narrowing type casting –** Narrowing type casting, also known as downcasting, is a data type conversion process used for –

* Narrowing a wider primitive type value to a smaller one

* Narrowing a superclass reference to its subclass reference

Example program of narrowing type casting

```
public class MyClass {

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

```
        double myDouble = 8.64;

        int myInt = (int) myDouble;
        System.out.println(myDouble);

        System.out.println(myInt);

    }

}
```

Output: 8.64

8

ii) Widening type casting – Widening type casting, also known as upcasting, is a data type conversion process taking place when -

* A subclass reference variable gets automatically placed within its superclass reference variable
* A small primitive type value gets automatically placed within a bigger primitive data type

```
public class MyClass {

  public static void main(String[] args) {

        int myInt = 6;

        double myDouble = myInt;

        System.out.println(myInt);

        System.out.println(myDouble);

  }

}
```

Output: 6

6.0

*** The ToString() and ToArray() methods –**

ToString() – toString() is a method to return an object's string representation. The Java compiler invokes the toString() method when an object is printed. This method is overridden to obtain the required output. Overriding the toString() method of an Object class is efficient for returning object values, and also reduces the coding one has to do

Code example of toString()

```
class Employee{
 int empcode;
String name;
String city;

 Student(int empcode, String name, String city){
this.empcode=empcode;
this.name=name;
this.city=city;
}
public String toString()
return empcode+' '+name+' '+city;
}
public static void main(String args[]){
Employee e1=new Employee(350,'Rohan','Kolkata');
Employee e2=new Employee(351,'Priya','Noida');

  System.out.println(e1);
 System.out.println(e2);
}
}
```

Output-

350 Rohan Kolkata

351 Priya Noida

toArray() -The toArray() method is used to fetch an array containing all the elements of an ArrayList object in an orderly sequence (arranged from first to last).

Code example of toArray()

```
import java.util.*;
public class test {
  public static void main(String[] args) {

   ArrayList<String> color_list = new ArrayList<String>(5);
   color_list.add('Red');
   color_list.add('Blue');
   color_list.add('Green');
   color_list.add('Yellow');
   color_list.add('Orange');

   System.out.println('Size of list: ' + color_list.size());
   for (String value : color_list) {
    System.out.println('Color = ' + value);
   }
   Object[] obj = color_list.toArray();
   System.out.println('Print elements from first to last:');
   for (Object value : obj) {
    System.out.println('Color = ' + value);
   }
  }
}
```

Output:

```
D:\java\javac test.java
D:\java\java test
Size of list: 5
Color = Red
Color = Blue
Color = Green
Color = Yellow
Color = Orange
Print elements from first to last:
Color = Red
Color = Blue
Color = Green
Color = Yellow
Color = Orange
```

Parsing

Parsing is a process by which a data type value is converted to another data type value. In Java, parsing is the conversion of a string to a primitive data type. It is an important process as string values cannot be used for arithmetic operations. The two types of parsing are top down and bottom up parsing.

Practical(60 minutes)

a) See the example programme of narrow typecasting below. Write a similar programme for narrowing down 10.46 and 7.89. Show the resulting output.

```
public class MyClass {
  public static void main(String[] args) {
        double myDouble = 8.64;
```

```
Int myInt = (int) myDouble;
        System.out.println(myDouble);
        System.out.println(myInt);
 }
}
```

b) See the example programme of wide typecasting below. Write a similar programme for widening the values 10.46 and 7.89. Show the resulting output.

```
public class MyClass {
 public static void main(String[] args) {
        int myInt = 6;
        double myDouble = myInt;
        System.out.println(myInt);
        System.out.println(myDouble);
 }
}
```

Instructions: The progress of students will be assessed with the exercises mentioned below.

MCQ

1. A Java variable refers to a container which holds a _____.

a) value

b) file system

c) file

d) None of the mentioned

2. _____ types are assigned to Java variables.

a) array

b) data

c) 30

d) 10

3. What keyword cannot be used to define a local variable ?

a) var

b) dynamic

c) static

d) None of the mentioned

4. What is used to store a local variable declaration statement?

a) a byte

b) a block

c) a bracket

d) None of the mentioned

5. What variable is created when an object is created?

a) static

b) local

c) instance

d) All of the mentioned

6. Static variable memory allocation only happens during the loading of a class in the _____.

a) directory

b) memory

c) array

d) None of the mentioned

7. What is the purpose of type casting in Java?

a) increasing the value of data

b) converting a variable type to another

c) initiating a new data instance

d) None of the mentioned

8. Narrow casting is also known as _____.

a) broadcasting

b) upcasting

c) downcasting

d) None of the mentioned

9. Narrow casting is used for narrowing a wider _____ data type value.

a) non-primitive

b) definitive

c) primitive

d) multiple

10. Widening type casting is used for -

a) larger to smaller data type casting

b) smaller to larger data type casting

c) small to smaller data type casting

d) None of the mentioned