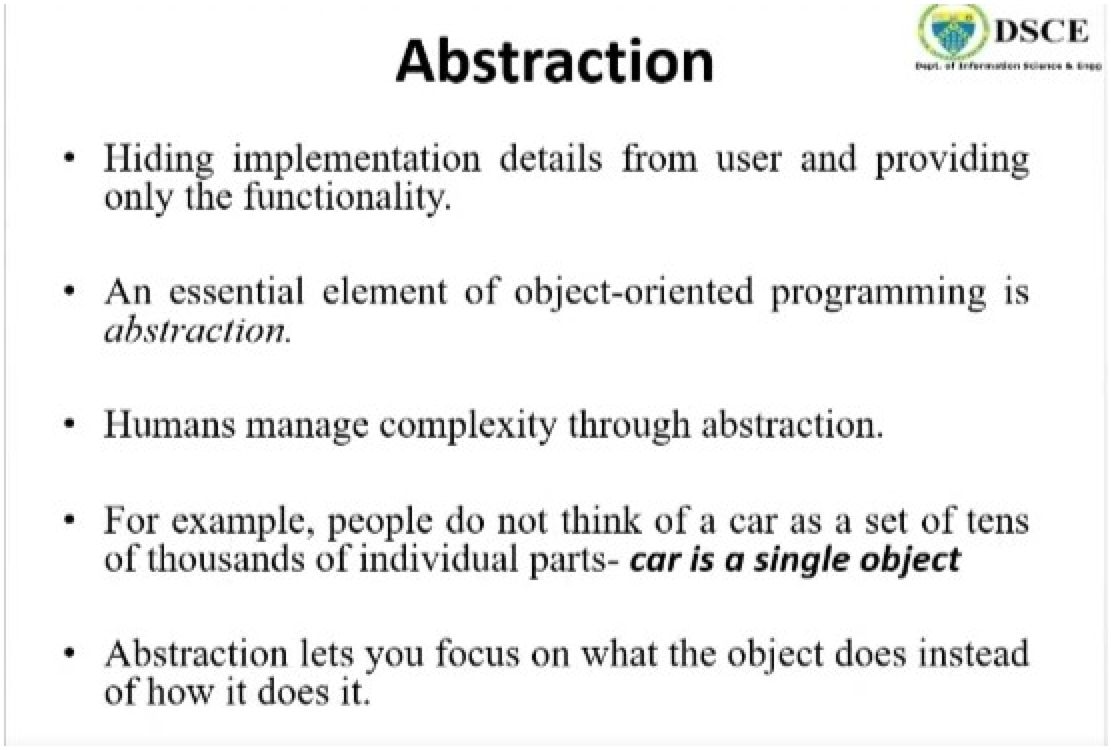
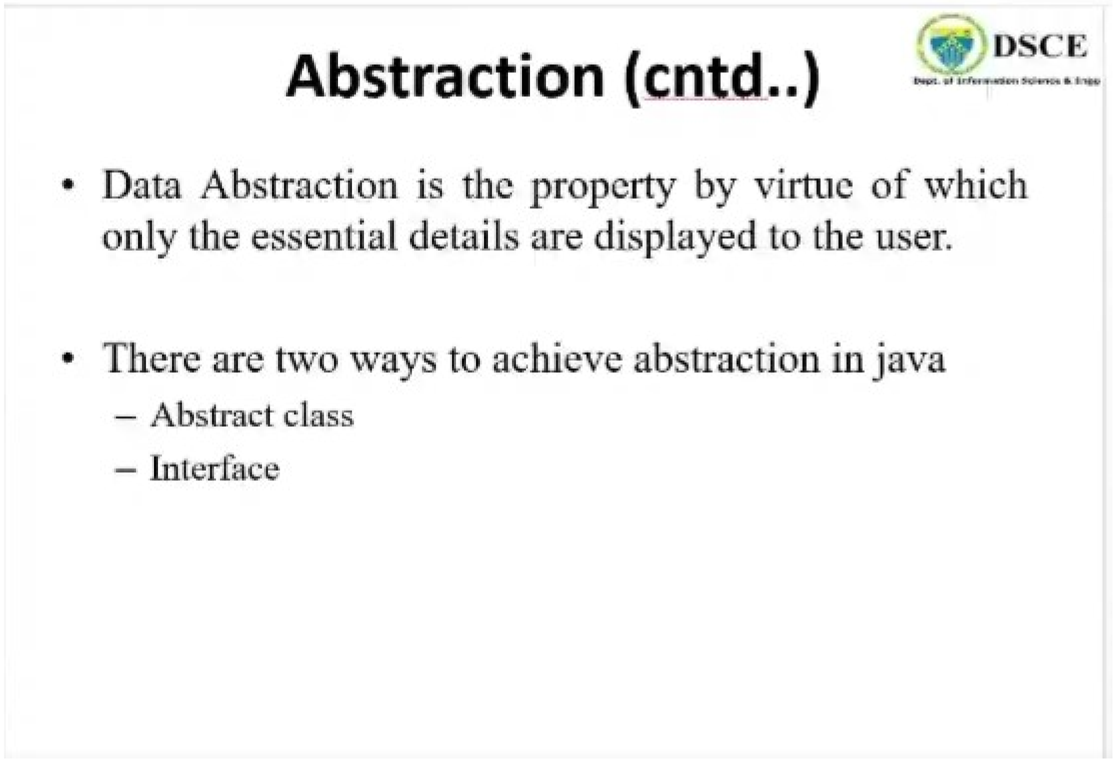


DSCE

Abstraction

- Hiding implementation details from user and providing only the functionality.
- An essential element of object-oriented programming is *abstraction.*
- Humans manage complexity through abstraction.
- For example, people do not think of a car as a set of tens of thousands of individual parts- ***car is a single object***
- Abstraction lets you focus on what the object does instead of how it does it.



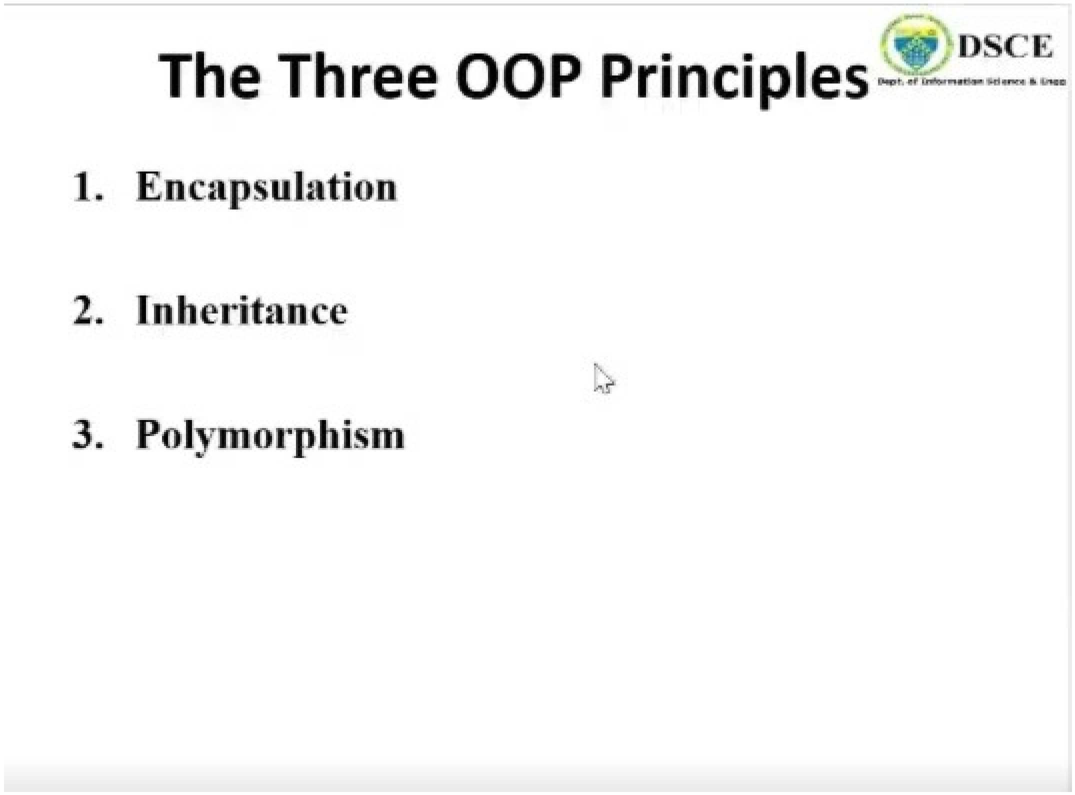
Abstraction (cntd..)

DSCE

Dept. of Information Science & Engg

- Data Abstraction is the property by virtue of which only the essential details are displayed to the user.

- There are two ways to achieve abstraction in java
  - Abstract class
  - Interface

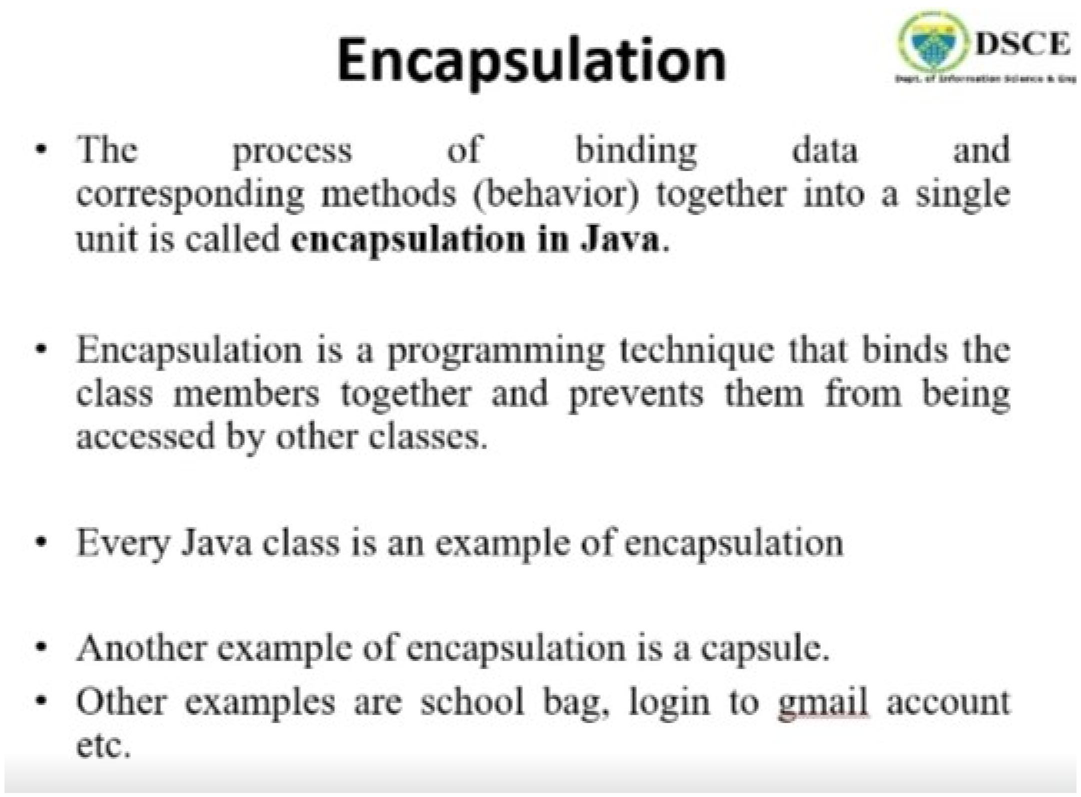


DSCE

Dept. of Information Science & Engg

The Three OOP Principles

1. **Encapsulation**
2. **Inheritance**
3. **Polymorphism**



Encapsulation

DSCE

- The process of binding data and corresponding methods (behavior) together into a single unit is called **encapsulation in Java**.

- Encapsulation is a programming technique that binds the class members together and prevents them from being accessed by other classes.

- Every Java class is an example of encapsulation

- Another example of encapsulation is a capsule.
- Other examples are school bag, login to gmail account etc.

Encapsulation(cntd..)

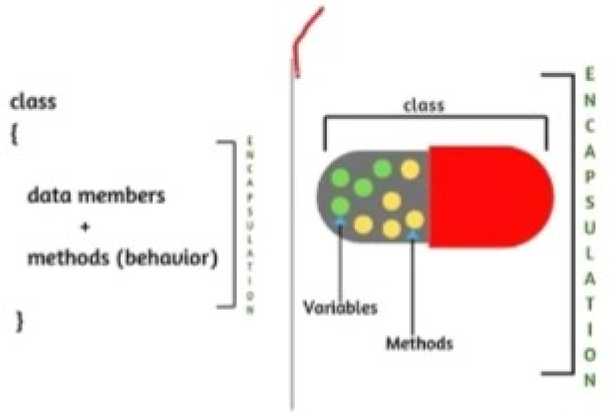
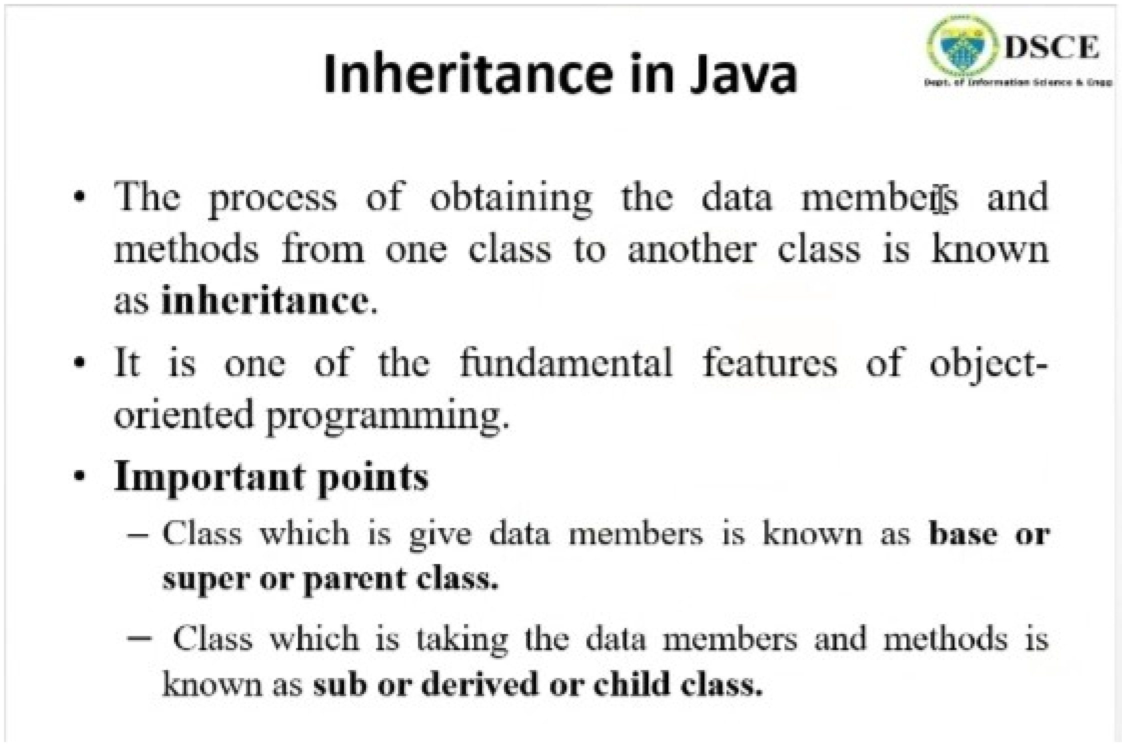


Fig: Encapsulation

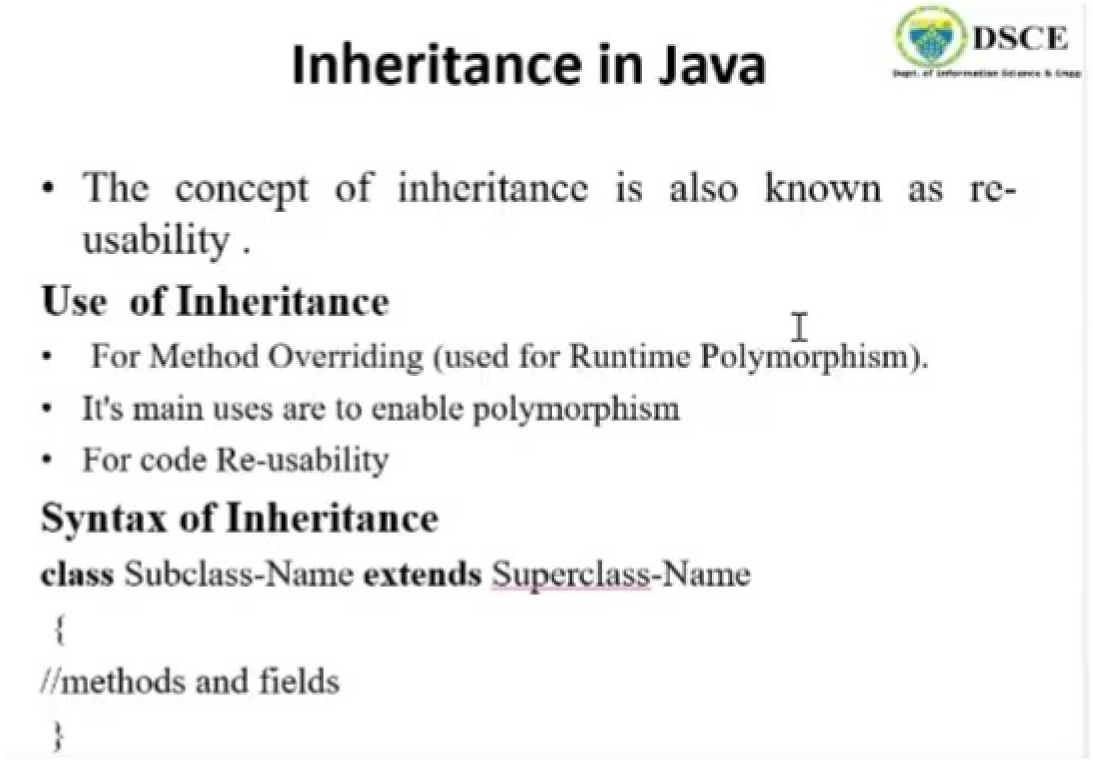


DSCE

Dept. of Information Science & Engg

Inheritance in Java

- The process of obtaining the data members and methods from one class to another class is known as **inheritance**.
- It is one of the fundamental features of object-oriented programming.
- **Important points**
  - Class which is give data members is known as **base or super or parent class.**
  - Class which is taking the data members and methods is known as **sub or derived or child class.**



Inheritance in Java

- The concept of inheritance is also known as re-usability .

Use of Inheritance

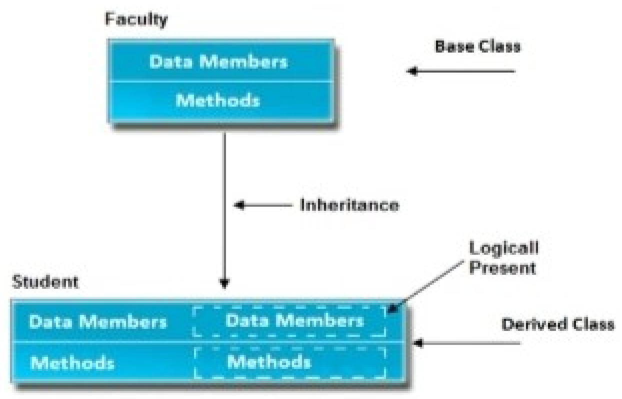
- For Method Overriding (used for Runtime Polymorphism).
- It's main uses are to enable polymorphism
- For code Re-usability

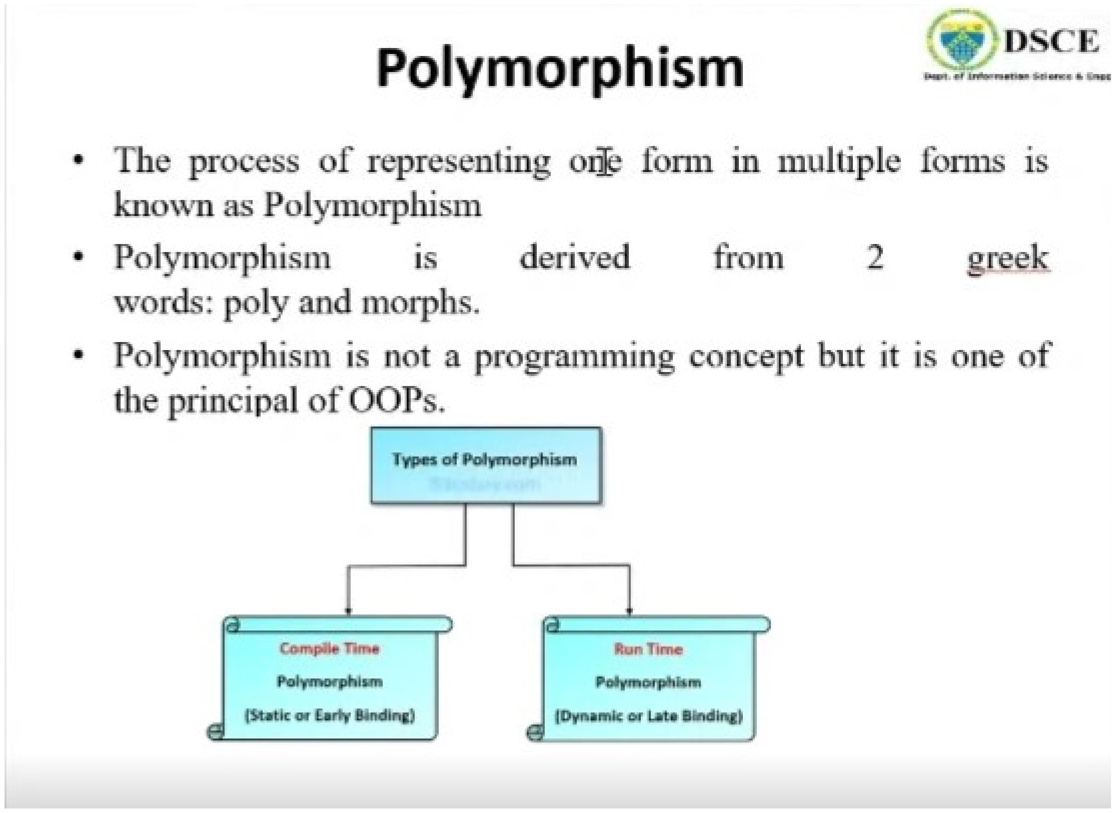
Syntax of Inheritance

```
class Subclass-Name extends Superclass-Name
 {
//methods and fields
 }
```

Inheritance in Java(cntd..)

- The real life example of inheritance is child and parents, all the properties of father are inherited by his son
- Another example is illustrated in diagram below

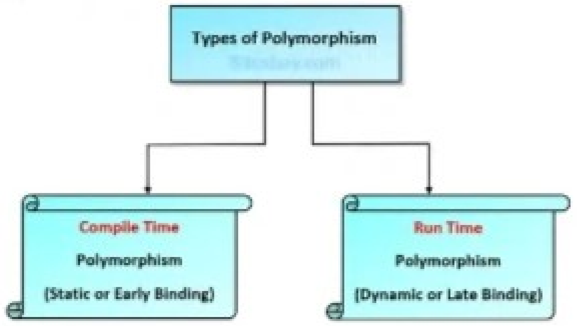




DSCE

Polymorphism

- The process of representing one form in multiple forms is known as Polymorphism
- Polymorphism is derived from 2 greek words: poly and morphs.
- Polymorphism is not a programming concept but it is one of the principal of OOPs.



Real life example of polymorphis in Java (cntd..)

DSCE

Dept. of Information Science & E

Real life example of polymorphism in Java

In Shopping malls behave like Customer

In Bus behave like Passenger

In School behave like Student

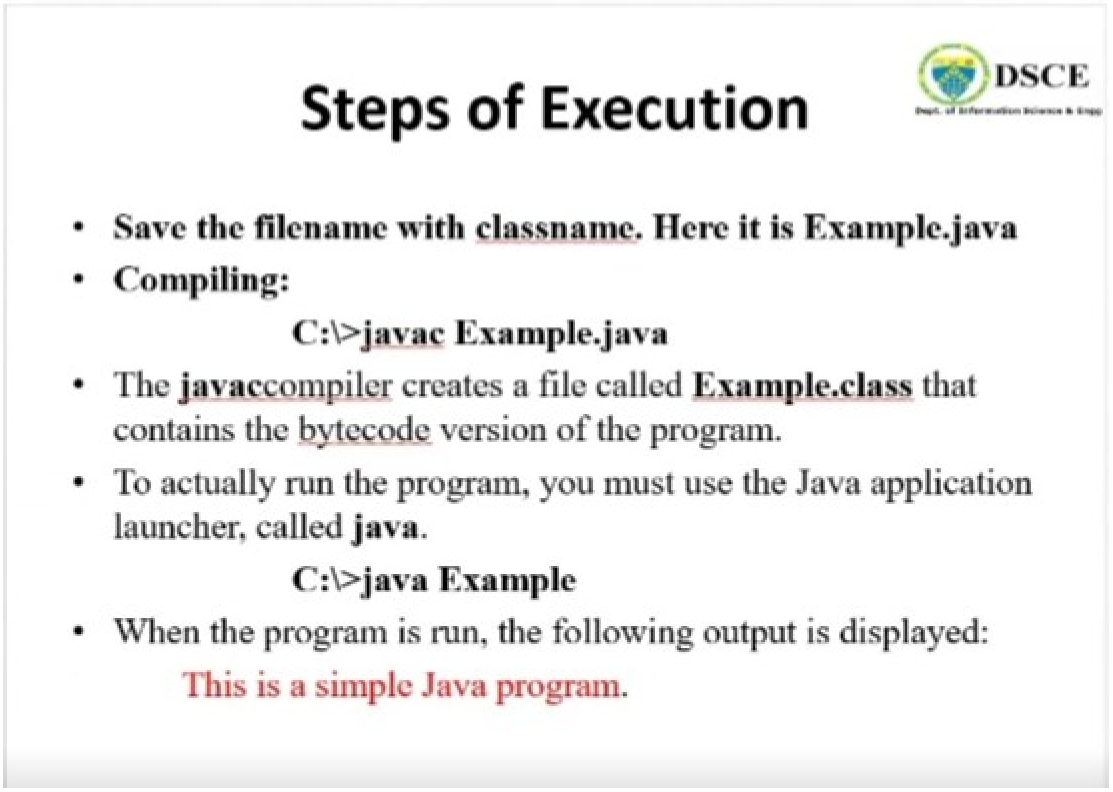
At Home behave like Son

A First Simple Program

DSCE

```
/*
This is a simple Java program. Call this file "Example.java".
*/

   class Example {
// Your program begins with a call to main(). public static
   void main(String args[]) {
System.out.println("This is a simple Java program.");
}
}
```



Steps of Execution

DSCE

- **Save the filename with classname. Here it is Example.java**
- **Compiling:**

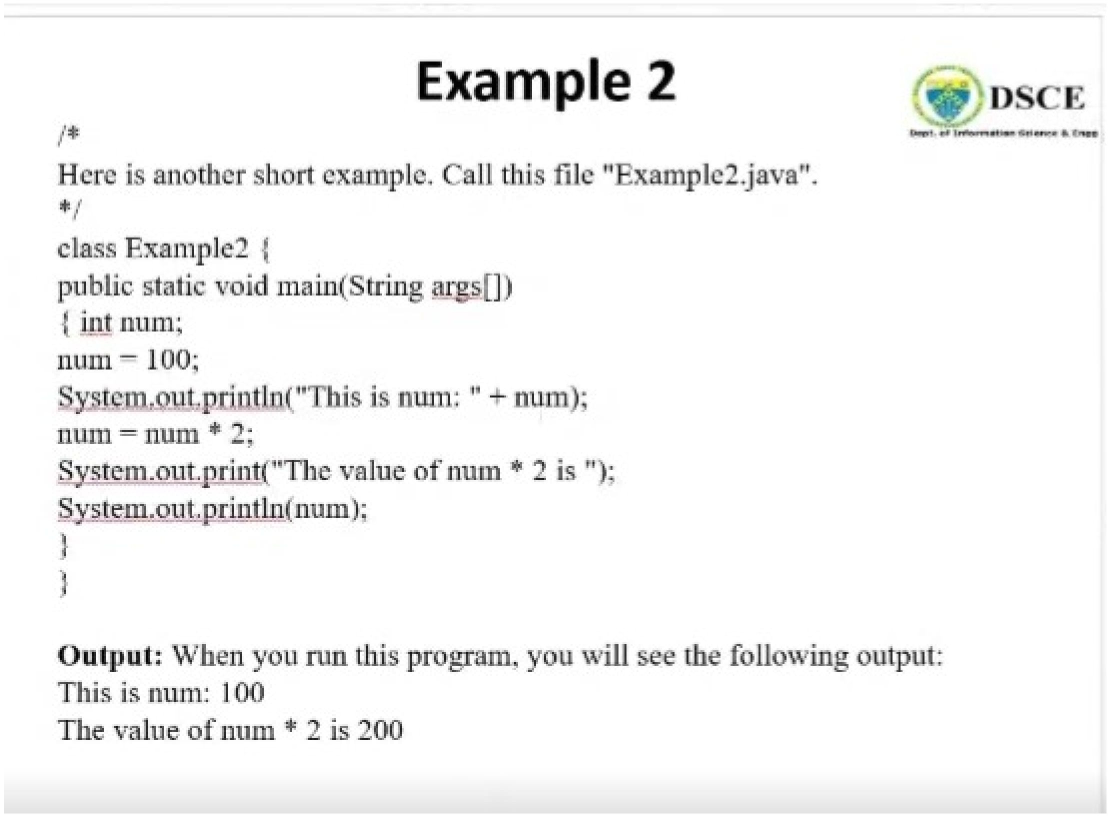
```
C:\>javac Example.java
```

- The **javac**compiler creates a file called **Example.class** that contains the bytecode version of the program.
- To actually run the program, you must use the Java application launcher, called **java**.

```
C:\>java Example
```

- When the program is run, the following output is displayed:

```
This is a simple Java program.
```



Example 2

```
/*
Here is another short example. Call this file "Example2.java".
*/
class Example2 {
public static void main(String args[])
{ int num;
num = 100;
System.out.println("This is num: " + num);
num = num * 2;
System.out.print("The value of num * 2 is ");
System.out.println(num);
}
}
```

Output: When you run this program, you will see the following output:

```
This is num: 100
The value of num * 2 is 200
```

Two Control Statements

DSCE

Dept. of Information Science & Engg

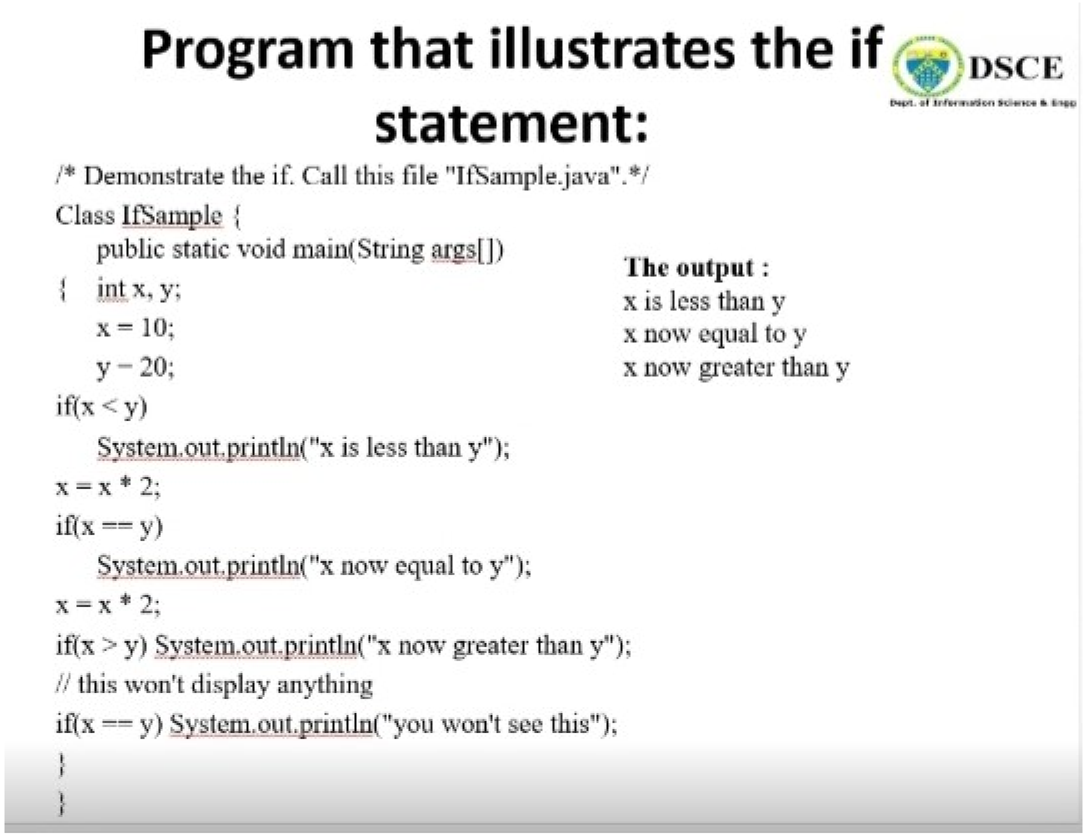
The if Statement

Syntax: **if(*condition*) *statement*;**

- Here, *condition* is a Boolean expression
- **Example:** if(num< 100)

```
System.out.println("num is less than 100");
```

Operator	Meaning
<	Less than
>	Greater than
==	Equal to



Program that illustrates the if statement:

DSCE
Dept. of Information Science & Engg

```
/* Demonstrate the if. Call this file "IfSample.java".*/
Class IfSample {
    public static void main(String args[])
{   int x, y;
    x = 10;
    y - 20;
if(x < y)
    System.out.println("x is less than y");
x = x * 2;
if(x == y)
    System.out.println("x now equal to y");
x = x * 2;
if(x > y) System.out.println("x now greater than y");
// this won't display anything
if(x == y) System.out.println("you won't see this");
}
}
```

The output :

```
x is less than y
x now equal to y
x now greater than y
```

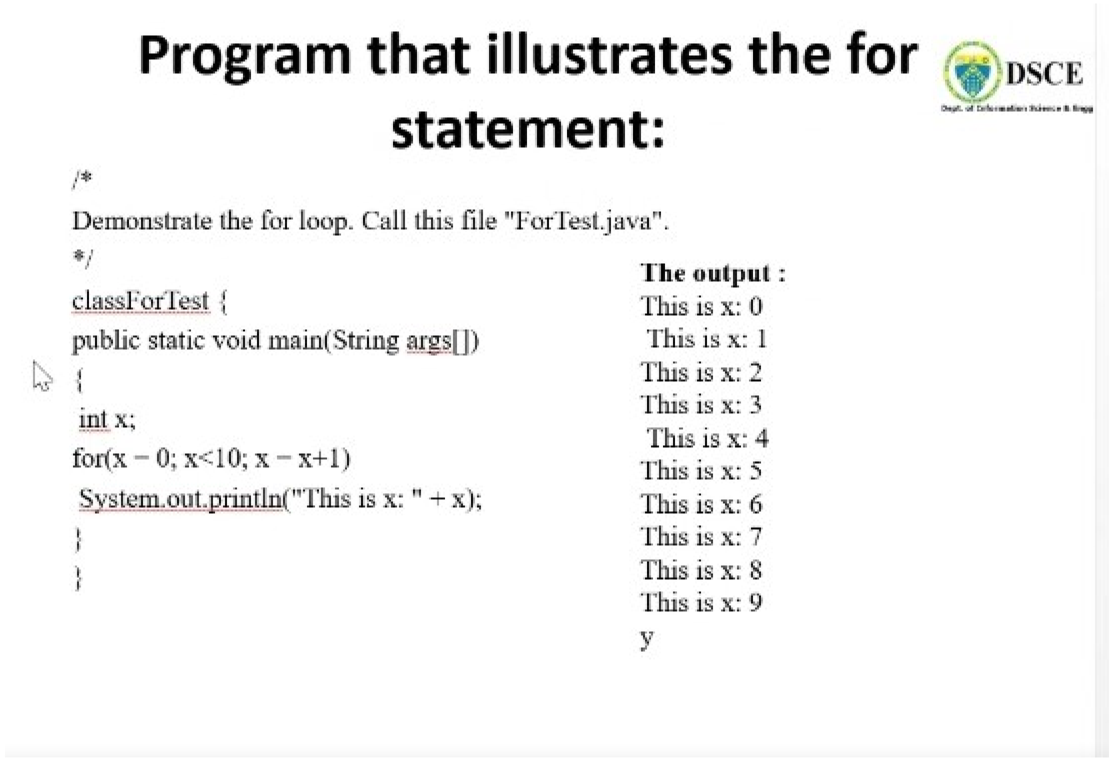
The for Loop

DSCE

- **Syntax:** i=0 i<10 i++

for(initialization; condition; iteration) statement;

- The initialization portion of the loop sets a loop control variable to an initial value.
- The condition is a Boolean expression that tests the loop control variable



Program that illustrates the for statement:

```
/*
Demonstrate the for loop. Call this file "ForTest.java".
*/
classForTest {
public static void main(String args[])
{
 int x;
for(x = 0; x<10; x = x+1)
 System.out.println("This is x: " + x);
}
}
```

The output :

```
This is x: 0
 This is x: 1
This is x: 2
This is x: 3
 This is x: 4
This is x: 5
This is x: 6
This is x: 7
This is x: 8
This is x: 9
y
```