



# Sri Indu College of Engineering & Technology :: Sheriguda (V), R.R.Dist

## “One Week on line Training Program on Skill Development on JAVA”

by Prof A K Jain

Date: 01-05-2021

Organized by S-Hub and P-Hub Cell

Days	Topic	Start Time	End Time	Duration (Minutes)	Participants
<b>Day - 1</b>	1. Introduction of JAVA. 2. Examples of JAVA programs	4/24/2021 18:23	4/24/2021 19:55	92	228
<b>Day - 2</b>	1. JAVA Key words 2. Online Compiling of JAVA	4/25/2021 18:19	4/25/2021 19:36	78	183
<b>Day - 3</b>	1. Theory of JAVA programming 2. Types of Classes 3. Examples of JAVA programs	4/26/2021 18:12	4/26/2021 19:43	91	125
<b>Day - 4</b>	1. Explanation of OOPS 2. Differences of functions and methods 3. Example programs are executed on online Compiler	4/27/2021 18:15	4/27/2021 20:01	107	312
<b>Day - 5</b>	1. Explanation of OOPS 2. Differences of functions and methods 3. Example programs are executed on online Compiler	4/28/2021 18:22	4/28/2021 19:52	90	155
<b>Day - 6</b>	1. Practice of JAVA programming 2. Example programs are executed on online Compiler	4/29/2021 18:19	4/29/2021 19:38	79	140
<b>Day - 7</b>	1. Practice of JAVA programming. 2. Example programs are executed on online Compiler	4/30/2021 18:22	4/30/2021 19:59	98	107

1<sup>st</sup> day class:

The screenshot shows a Zoom meeting window with a grid of 12 participant tiles. The top row includes tiles for 'SICET (Autonomous)', 'Parushu' (highlighted with a green border), 'lenovo', and a tile for '18D41A04F0 A...'. The second row has '20D45A0243 in...', '17D41A0271', a tile with a large 'B' (ID: 18D41A04D5, B...), and '18D41A04C1 ha...'. The third row shows '18D41A04D9 M...', '18D41A04A6 N...', 'Jay surya 4A1', and '18D41A04H7'. The bottom row includes 'Arun Kumar L4...', '18D41A0425 B...', '18D41A0405 Ar...', and '18D41A0243'. The bottom toolbar contains icons for Mute, Start Video, Security, Participants (39), Share Screen, Reactions, More (2), and End. The right sidebar shows a list of 38 participants, including Parushu (Host, me), lenovo (Co-host), and several other users with their IDs and muted status.

The screenshot shows a Zoom meeting window displaying a slide titled 'Recording...' with the following content:

- Step 1** Write source code
- Step 2** Compile (Translate) source code into machine code
- Step 3** Execute (Run) machine code

The slide features a flowchart of the Java compilation process:

```
graph TD; A[Source code (Xxx.java)] --> B[Java Compiler  
javac Xxx.java]; B --> C[Java Bytecode (Xxx.class)]; C --> D[Java Runtime  
java Xxx]; E[Input] --> D; D --> F[Output];
```

The flowchart includes a warning box: 'Please move this window away from the compiler'. The right sidebar shows a list of 113 participants, including Parushu (Host, me), lenovo (Co-host), and 17-4E3 HAKEEM.

Zoom Meeting | You are viewing lenovo's screen | View Options

Parushu | lenovo | 18D41A04M9

Participants (124)

Find a participant

- 1H 18D41A04C1 haripriya
- B 18D41A04D5. Bhanusha .
- 18D41A04D8 M. Sravani
- 18D41A04F0 Akash
- 18D41A04F4 .
- 1 18D41A04F8
- 1V 18D41A04F9 Vivek
- 18D41A04G2 Vikas
- 1 18D41A04G9

Invite | Mute All

Chat

```

1 public class MyClass {
2     public static void main(String args[]) {
3         int original_balance=10000;
4         int deposit =2500;
5         int updated_balance = original_balance +5;
6         int money;
7         int k= y - k ;
8
9         System.out.println("Sum of x+y = " + z);
10        System.out.println("diff is"+ kk);
11    }
12 }

```

Execute Mode, Version, Inputs & Arguments

JDK 11.0.4 | Interactive | CommandLine Arguments

Execute

Result  
compiled and executed in 0.924 secs

```

Sum of x+y = 26335
diff is24910

```

Notes

Unmute | Start Video | Security | Participants | Share Screen | Reactions | More | End

7:32 PM 4/24/2021

Zoom Meeting | Recording... | View

lenovo | Parushu | 17-4E3 HAKEEM | 17D41A04K6 | 18D41A04M9

17D41A0271 | 18D41A04C1 ha...

Vineetha | 18D41A04G9 | 17D41A0414 | 18D41A04A3. L... | Ask to Unmute | CORONAVIRUS

18D41A0433 va... | 18D41A04L9 | 17D41A04E0 | 17D41A04L3 | 18D41A04N4(K...

Mute | Start Video | Security | Participants | Chat | Share Screen | Pause/Stop Recording | Reactions | More | End

7:50 PM 4/24/2021

Zoom Meeting

Recording... View

17-4E3 HAKEEM	Parushu	18D41A04M9	M	17D41A0409 ...	18D41A04 C5
17D41A04E2 Na...	18D41A0432	18D45A0401 ...	18D41A04F0 ...	17D41A0271	18D41A04C1 ha...
18D41A04A6 ...	Vineetha	18D41A04G9	17D41A0414	18D41A04A3. L...	19D45A0412 ...
18D41A0433 va...	19D45A0405 ...	17D41A04E0	18D41A04N4(K...	18D41A0429 So...	B.Anusha 17D4...

Mute Start Video Security Participants 75 Chat 2 Share Screen Pause/Stop Recording Reactions More End

7:52 PM 4/24/2021

Zoom Meeting

Recording... View

17-4E3 HAKEEM	Parushu	18D41A04M9	M	17D41A0409 ...	18D41A04 C5
17D41A04E2 Na...	18D41A0432	18D45A0401 ...	17d-4G6/venka...	18D41A04F0 ...	17D41A0271
18D41A04C1 ha...	18D41A04A6 ...	Vineetha	18D41A04G9	17D41A0414	18D41A04A3. L...
19D45A0412 ...	18D41A0433 va...	19D45A0405 ...	17D41A04E0	18D41A04N4(K...	18D41A0429 So...

Mute Stop Video Security Participants 71 Chat 2 Share Screen Pause/Stop Recording Reactions More End

7:53 PM 4/24/2021

2<sup>nd</sup> class:

Zoom Meeting

Participants (101)

Find a participant

- 1 19D45A0422
- G G.Jyothika/17D41A0470
- G.G.Pramod 471
- P PRAVEEN
- S Srilaya 17D41A0433
- S Sumana17D41A0435
- SL Swapna L404
- V venkat
- V Vineetha
- 1S 18D41A0429 Sowmya

Go! It! Mute All

6:37 PM 4/25/2021

Zoom Meeting

You are viewing lenovo's screen

Participants (105)

Find a participant

```
Project Name: AccountDetails.java
1 import java.util.Date;
2 import java.lang.*;
3
4 public class AccountDetails {
5     public static void main(String args[]) {
6         String name = "K K JAIN";
7
8         // Show date and time using only Date methods.
9
10
11 Date date = new Date();
12 // display time and date using toString()
13 System.out.println(date);
14 // Display number of milliseconds since midnight, January 1, 1970 GMT
15 long msec = date.getTime();
16 System.out.println("Milliseconds since Jan. 1, 1970 GMT = " + msec);
17
18
19 int original_balance=10000;
20 System.out.println("Hello Mr. "+name);
21 System.out.println("Original Balance is Rs." + original_balance);
22
23 int deposit =2000;
24 System.out.println("You have deposited Rs. "+deposit);
25 int updated_balance = original_balance + deposit;
26 System.out.println("After deposit the new balance is Rs." + updated_balance);
27
28 int withdrawal = 500;
29 System.out.println("You have withdrawal Rs. "+withdrawal);
30 int updated_withdrawal = updated_balance - withdrawal ;
31 System.out.println("Current balance is Rs." + updated_withdrawal);
32
33 }
```

Result

6:42 PM 4/25/2021

Zoom Meeting | You are viewing lenovo's screen | View Options

Participants (110)

Find a participant

- G.Pramod 471
- K.kirankumar/17D41A04A6
- K.Poojitha/17D41A0490
- PRAVEEN
- Srilaya 17D41A0433
- Sumana17D41A0435
- SL Swapna L404
- venkat
- Vineetha
- 1A 18D41A0489 AJITH REDDY

Unmute | Start Video | Security | Participants | Share Screen | Reactions | More | End

6:52 PM 4/25/2021

```

Project Name: AccountDetails.java
1 import java.util.Date;
2 import java.lang.*;
3
4 public class AccountDetails {
5     public static void main(String args[]) {
6         String name = "A C JALVI";
7
8         // Show date and time using only Date methods.
9
10        Date date = new Date();
11        // display time and date using toString()
12        System.out.println(date);
13        // Display number of milliseconds since midnight, January 1, 1970 GMT
14        long msec = date.getTime();
15        System.out.println("Milliseconds since Jan. 1, 1970 GMT = " + msec);
16
17
18
19        int original_balance=10000;
20        System.out.println("Hello Mr. "+name);
21        System.out.println("Original Balance is Rs."+original_balance);
22
23        int deposit =2500;
24        System.out.println("You have deposited Rs."+"deposit");
25        int updated_balance = original_balance + deposit;
26        System.out.println("After deposit the new balance is Rs." + updated_balance);
27
28        int withdrawal = 500;
29        System.out.println("You have withdrawl Rs."+"withdrawal");
30        int updated_withdrawl = updated_balance - withdrawal;
31        System.out.println("Current balance is Rs." + updated_withdrawal);
32    }
33 }
  
```

Execute Mode, Version, Inputs & Arguments

CommandLine Arguments

JDK 11.0.4

Interactive

Execute

Result

compiled and executed in 0.854 sec(s)

```

Sun Apr 25 13:12:58 GMT 2021
Milliseconds since Jan. 1, 1970 GMT = 1619356378389
Hello Mr. A C JALVI
Original Balance is Rs.10000
You have deposited Rs.2500
After deposit the new balance is Rs.12500
You have withdrawl Rs.500
Current balance is Rs.12000
  
```

3<sup>rd</sup> day 26-04-2021

Zoom Meeting | You are viewing lenovo's screen | View Options

Participants (53)

Find a participant

- 1 17D41A04H1
- 1 17D41A04H6
- 17D41A04H8 Ajay
- 17D41A04J4 .
- 17D41A04J6 - Sri Charan
- 1 17D41A04K5
- 1 17D41A04L0
- 1 17D41A04M1
- 1 17D41A04N0
- 1 18D41A0231

Mute | Start Video | Security | Participants | Share Screen | Reactions | More | End

6:38 PM 4/26/2021

### Characteristics of Java

- Java Is Simple
- Java Is Object-Oriented
- Java Is Distributed
- Java Is Interpreted
- Java Is Robust
- Java Is Secure
- Java Is Portable
- Java's Performance
- Java Is Multithreaded

Zoom Meeting | You are viewing lenovo's screen | View Options

Parushu | 18D41A04M9

## Calling a Method

- If the method returns a value
  - `int larger = max(3, 4);`
- If the method returns void, a call to the method must be a statement.
  - `System.out.println("Welcome to Java!");`

Recording... | Unmute | Start Video | Security | Participants (95) | Share Screen | Reactions | More | End

Participants (95)

Find a participant

- Parushu (Host, me)
- lenovo (Co-host)
- 17-4F6 /Nenavath Praveen
- 17BE1A0537 Swetha
- 17D1A04E9
- 17D41A0409 Rohith
- 17D41A0413
- 17D41A0414
- 17D41A0434 Gayathri
- 17D41A0444 ROHINI

6:44 PM 4/27/2021

Zoom Meeting | You are viewing lenovo's screen | View Options

Parushu | 17D41A0458Sre... | lenovo

## A Simple Java Program

```
// My First Program!!
// name of file = Helloworld.java
public class Helloworld {
    public static void main(String[] args){
        system.out.println("Hello Friends, welcome
        to Basics of Java!");
    }
}
```

Recording... | Unmute | Start Video | Security | Participants (57) | Share Screen | Reactions | More | End

Participants (57)

Find a participant

- 17D41A04A7
- 17D41A04D3
- 17D41A04E0
- 17D41A04G7
- 17D41A04H1
- 17D41A04H6
- 17D41A04H8 Ajay
- 17D41A04J4.
- 17D41A04J6 - Sri Charan
- 17D41A04K5

6:41 PM 4/26/2021

Zoom Meeting | You are viewing lenovo's screen | View Options

Parushu | lenovo | 18D41A04M4 R... | View

Recording... ||

## Primitive Data Types

1) boolean	Boolean	} Wrapper Class
2) byte	Byte	
3) short	Short	
4) int	Int	
5) long	Long	
6) float	Float	
7) double	Double	
8) char	Char	

Unmute My Audio (Alt+A). Or you can simply press and hold the space bar to temporarily unmute.

Participants (74)

Find a participant

- Parushu (Host, me)
- lenovo
- 17D41A04M1
- 17-4F6 /Nenavath Praveen
- 17D41A0212 Dinesh CHERUPAL...
- 17D41A0409 Rohith
- 17D41A0413
- 17D41A0430
- 17D41A0442 poojitha
- 17D41A0446 Uma

Unmute Start Video Security Participants Share Screen Reactions More End

7:05 PM 4/26/2021

Zoom Meeting | You are viewing lenovo's screen | View Options

Parushu | lenovo | 17D41A04J6 - Sri... | View

Recording... ||

## Benefits of Methods

- Write a method once and reuse it anywhere.
- Information hiding: Hide the implementation from the user.
- Reduce complexity.

Participants (68)

Find a participant

- Parushu (Host, me)
- lenovo
- 17D41A04J6 - Sri Charan
- 17-4F6 /Nenavath Praveen
- 17D41A0212 Dinesh CHERUPAL...
- 17D41A0413
- 17D41A0430
- 17D41A0441Vaishnavi
- 17D41A0442 poojitha
- 17D41A0446 Uma

Unmute Start Video Security Participants Share Screen Reactions More End

7:35 PM 4/26/2021



4<sup>TH</sup> DAY 27-04-2021

Zoom Meeting | You are viewing lenovo's screen | View Options

Participants: Parushu, 17D41A04M1, lenovo

### Example

```
public class TestMethodOverloading {  
    /** Main method */  
    public static void main(String[] args) {  
        // Invoke the max method with int parameters  
        System.out.println("The maximum between 3 and 4 is "+ max(3, 4));  
        // Invoke the max method with the double parameters  
        System.out.println("The maximum between 3.0 and 5.4 is"+ max(3.0,5.4));  
    }  
    /** Return the max between two int values */  
    public static int max(int num1, int num2) {  
        if (num1 > num2)  
            return num1;  
        else  
            return num2;  
    }  
    /** Find the max between two double values */  
    public static double max(double num1, double num2) {  
        if (num1 > num2)  
            return num1;  
        else  
            return num2;  
    }  
}
```

Participants (130)

Find a participant

- 1 17D41A0582
- 1 17D41A0594
- 1 17D41A05C7
- 1 17D41A05E5
- 1 17D41A... Ask to Unmute More >
- 1 17D41A05G5
- P 17D41A05H1
- 1 17D41A05H3
- 1 17D41A05M1
- 17D41A05N8

Unmute Start Video Security Participants 130 Share Screen Reactions More End

6:50 PM 4/27/2021

Zoom Meeting | You are viewing lenovo's screen | View Options

Participants: Parushu, 17D41A04M1, 17D41A04J6...

### Example

```
public class TestMethodOverloading {  
    /** Main method */  
    public static void main(String[] args) {  
        // Invoke the max method with int parameters  
        System.out.println("The maximum between 3 and 4 is "+ max(3, 4));  
        // Invoke the max method with the double parameters  
        System.out.println("The maximum between 3.0 and 5.4 is"+ max(3.0,5.4));  
    }  
    /** Return the max between two int values */  
    public static int max(int num1, int num2) {  
        if (num1 > num2)  
            return num1;  
        else  
            return num2;  
    }  
    /** Find the max between two double values */  
    public static double max(double num1, double num2) {  
        if (num1 > num2)  
            return num1;  
        else  
            return num2;  
    }  
}
```

Participants (166)

Find a participant

- 15 17d41a0 589
- 17D41A0409 Rohith
- 1 17D41A0413
- 17D41A0414
- 1 17D41A0436
- 1 17D41A0441
- 1R 17D41A0444 ROHINI
- 1U 17D41A0446 Uma
- 1 17D41A0449
- 1 17D41A... Ask to Unmute More >

Unmute Start Video Security Participants 166 Share Screen Reactions More End

6:53 PM 4/27/2021

Zoom Meeting | You are viewing lenovo's screen | View Options

Parushu | 17D41A04M1 | lenovo

Recording... ||

## Scope of Local Variables

- A **local variable**: a variable defined inside a method.
- **Scope**: the part of the program where the variable can be referenced.
- The scope of a local variable starts from its declaration and continues to the end of the block that contains the variable. A local variable must be declared before it can be used.

Participants (177)

Q Find a participant

- L lenovo (Co-host)
- 1/ 17-4F6 /Nenavath Praveen
- 1 17D1A04E9
- 15 17d41a0 589
- 17D41A0409 Rohith
- 1 17D41A0413
- 17D41A0414
- 1 17D41A0436
- 1 17D41A0441
- 1R 17D41A0444 ROHINI

Unmute | Start Video | Security | Participants | Share Screen | Reactions | More | End

6:59 PM 4/27/2021

Zoom Meeting | You are viewing lenovo's screen | View Options

Parushu | 17D41A04M1

Recording... ||

## What Is an Object?

An object is a software bundle of related **state** and **behavior**.  
Software objects are often used to model the real-world objects that we find in everyday life.

## Classes & Objects in Java

Class	Car
<b>Data Members</b>	model color brand
<b>Methods</b>	speed() size()

model - Ertiga  
color - Mehroon  
brand - Maruti

model - XUV500  
color - Black  
brand - Mahindra

model - Swift  
color - Red  
brand - Maruti

## What Is a Class?

A class is a blueprint or prototype from which objects are created.

class | objects

Car | Audi | Nissan | Volvo

Participants (136)

Q Find a participant

- P Parushu (Host, me)
- L lenovo (Co-host)
- 1 17D41A04M1
- ( 17D41A05H7)
- 1/ 17-4F6 /Nenavath Praveen
- 1 17D1A04E9
- 17D41A0430 B.Anvesh
- 1 17D41A0436
- 1R 17D41A0444 ROHINI
- 1U 17D41A0446 Uma

Unmute | Start Video | Security | Participants | Share Screen | Reactions | More | End

7:30 PM 4/27/2021

Zoom Meeting

Parushu 17D41A04M1 lenovo

Recording...

## CLASS VERSUS OBJECT

CLASS	OBJECT
A template for creating or instantiating objects within a program	An instance of a class
Logical entity	Physical entity
Declared with the "class" keyword	Created using the "new" keyword
A class does not get any memory when it is created	Objects get memory when they are created
A class is declared once	Multiple objects are created using a class

Participants (124)

Find a participant

- Parushu (Host, me)
- lenovo (Co-host)
- (17D41A05H7)
- 17-4F6 /Nenavath Praveen
- 17D1A04E9
- 17D41A0430 B.Anvesh
- 17D41A0436
- 17D41A0444 ROHINI
- 17D41A0446 Uma
- 17D41A0449

7:41 PM 4/27/2021

5<sup>th</sup> day 28-04-2021

Zoom Meeting

Participants (48)

Find a participant

Parushu (Host, me)

17-4F6 /Nenavath Praveen

17D41A0409 Rohith

17D41A0430 B.Anvesh

17D41A0483

17D41A04A7

17D41A04E0

17D41A04E9

17D41A04J6 - Sri Charan

17D41A04K5

6:35 PM 4/28/2021

sathya Internet access

18D41A04A1 K...

17D41A04J6 - S...

18D41A04A3.le...

18d41a0575

18D41A05H0 C...

18D41A0465 E...

17D41A0575

19D45A0407 S...

chetana

18D41A05K7 N...

18D41A04M9

NIHARIKA 415

19D45A0412 R...

18D41A0231

Zoom Meeting

Parushu

19D45A0407 ...

chetana

Participants (58)

Find a participant

- Parushu (Host, me)
- lenovo (Co-host)
- 17-4F6 /Nenavath Praveen
- 17D41A0409 Rohith
- 17D41A0413
- 17D41A0430
- 17D41A0483
- 17D41A04A7
- 17D41A04E0
- 17D41A04E9

6:38 PM 4/28/2021

Zoom Meeting

Parushu

18D41A04M9

17D41A04J6 ...

Participants (80)

Find a participant

- Parushu (Host, me)
- lenovo (Co-host)
- (17D41A05H7)
- 17-4F6 /Nenavath Praveen
- 17D41A0409 Rohith
- 17D41A0413
- 17D41A0430
- 17D41A0441 vaishnavi
- 17D41A0444 ROHINI
- 17D41A0449 Chaitanya Shree

6:58 PM 4/28/2021

Recording...

jdoodle.com/online-java-compiler/#

Project Name: **BoxDemo.java**

```

1 - /* A program that uses the Box class.
2   Call this file BoxDemo.java
3   */
4 - class Box {
5     double width;
6     double height;
7     double depth;
8     } // class Box ends here
9
10 // This class declares an object of type Box
11
12 - public class BoxDemo {
13     public static void main(String args[]) {
14         Box mybox = new Box(); // new keyword creates an object = mybox here
15         double vol;
16
17         // assign values to mybox's instance variables
18         mybox.width = 10;
19         mybox.height = 20;
20         mybox.depth = 15;
21
22         // compute volume of box
23
24         vol = mybox.width * mybox.height * mybox.depth;
25         System.out.println("Volume is " + vol);
26     } // end of main
27 } // end of BoxDemo class

```

Volume is 3000.00

Zoom Meeting

Parushu 18D41A04M9

Recording... joodle.com/online-java-compiler/#

Project Name: **BoxDemo.java**

```

1 /* A program that uses the Box class.
2 Call this file BoxDemo.java
3 */
4 class Box {
5     double width;
6     double height;
7     double depth;
8 } // class Box ends here
9
10 // This class declares an object of type Box
11
12 public class BoxDemo {
13     public static void main(String args[]) {
14         Box mybox = new Box(); // new keyword creates an object = mybox here
15         double vol;
16
17         // assign values to mybox's instance variables
18         mybox.width = 10;
19         mybox.height = 20;
20         mybox.depth = 15;
21
22         // compute volume of box
23
24         vol = mybox.width * mybox.height * mybox.depth;
25         System.out.println("Volume is " + vol);
26     } // end of main
27 } // end of BoxDemo class

```

Volume is 3000.00

Participants (80)

Find a participant

- Parushu (Host, me)
- lenovo (Co-host)
- (17D41A05H7)
- 17-4F6 /Nenavath Praveen
- 17D41A0409 Rohith
- 17D41A0413
- 17D41A0430
- 17D41A0441 vaishnavi
- 17D41A0444 ROHINI
- 17D41A0449 Chaitanya Shree

6:58 PM 4/28/2021

6th day 29-04-2021

Zoom Meeting

You are viewing lenovo's screen

Parushu lenovo 18D41A04F5

Recording... Objects and Classes in java 29 04 21 - Microsoft PowerPoint

Project Name: **ConstructorBoxDemo.java**

```

1 /* Here, Box uses a constructor to initialize the
2 dimensions of a box.
3 */
4 class Box {
5     double width;
6     double height;
7     double depth;
8     // This is the constructor for Box.
9     Box() {
10        System.out.println("Constructing Box");
11        width = 100.56;
12        height = 110.87;
13        depth = 0.810;
14    }
15    // compute and return volume
16    double volume() {
17        return width * height * depth;
18    }
19 }
20
21 public class ConstructorBoxDemo {
22     public static void main(String args[]) {
23
24         // declare, allocate, and initialize Box objects
25         Box mybox1 = new Box();
26         Box mybox2 = new Box();
27         double vol;
28
29         // get volume of first box
30         vol1 = mybox1.volume();
31         System.out.println("Volume is " + vol);
32         // get volume of second box
33         vol1 = mybox2.volume();
34         System.out.println("Volume is " + vol);
35     }
36 }

```

Constructing Box  
Constructing Box  
Volume is 98657.16991200001  
Volume is 98657.16991200001

Participants (50)

Find a participant

- Parushu (Host, me)
- lenovo (Co-host)
- 17D41A0409 Rohith
- 17D41A0413
- H 17D41A0464
- 1 17D41A04E9
- 1 17D41A04K5
- 1 17D41A04L0
- 1 17D41A04L3
- 1 17D41A0515

6:37 PM 4/29/2021

Recording... You are viewing lenovo's screen View Options

Home Insert Design Animations Slide Show Review View MathType

Clipboard Paste Copy Format Painter New Slide Delete Slides Font Paragraph Drawing Editing

Slides Outline

### Exception Handling

An *exception* is an *abnormal* condition that arises in a code sequence at run time. In other words, **an exception is a runtime error**. In computer languages that do not support exception handling, errors must be checked and handled manually—typically through the use of error codes, and so on.

This approach is as cumbersome as it is troublesome. Java's exception handling avoids these problems and, in the process, brings run-time error management into the object-oriented world.

Unmute Start Video Security Participants Chat Share Screen Pause/Stop Recording Reactions More End

Zoom Meeting

Parushu lenovo 18D41A04M9

Participants (61)

Find a participant

- Parushu (Host, me)
- lenovo (Co-host)
- 17D41A0409 Rohith
- 17D41A0413
- 17D41A... Ask to Unmute More >
- 17D41A0464
- 17D41A04D0 - varshith
- 17D41A04E0 VAMSHI
- 17D41A04E9
- 17D41A04F8

Recording... Objects and Classy in Java 20 04 21 - Microsoft PowerPoint

Clipboard Paste Copy Format Painter New Slide Delete Slides Font Paragraph Drawing Editing

Slides Outline

```

java.lang.Object
├── java.lang.Throwable
│   ├── java.lang.Exception
│   │   ├── java.io.IOException
│   │   └── java.lang.RuntimeException
│   └── java.lang.ArithmeticException
├── java.lang.IndexOutOfBoundsException
├── java.lang.ArrayIndexOutOfBoundsException
├── java.util.NoSuchElementException
├── java.util.InputMismatchException
├── Others...
├── java.lang.Error
├── java.lang.VirtualMachineError
├── java.lang.OutOfMemoryError
├── java.lang.InternalError
└── Others...
    
```

Click to add notes

Slide 18 of 29 Office Theme English (India) 6:47 PM 4/29/2021

Zoom Meeting

Parushu lenovo 18D41A04M9

Recording... Design Animations Slide Show Review View MultiType

```

import java.util.Scanner;
public class DivisionMistakeCaught
{
    public static void main(String[] args)
    {
        Scanner input = new Scanner(System.in);
        int numerator, denominator, result;
        System.out.println("Enter numerator >> ");
        numerator = input.nextInt();
        System.out.println("Enter denominator >> ");
        denominator = input.nextInt();
        try
        {
            result = numerator / denominator;
            System.out.println(numerator + " / " + denominator +
                " = " + result);
        }
        catch(ArithmeticException mistake)
        {
            System.out.println("Arithmetic exception was thrown and caught");
        }
        System.out.println("End of program");
    }
}

```

Participants (65)

Find a participant

- Parushu (Host, me)
- lenovo (Co-host)
- 17D41A0409 Rohith
- 17D41A0413
- 17D41A0436
- 17D41A0464
- 17D41A0495 Sravan
- 17D41A04D0 - varshith
- 17D41A04E0 VAMSHI
- 17D41A04E9

6:55 PM 4/29/2021

7<sup>th</sup> day 30-04-2021

Zoom Meeting

You are viewing lenovo's screen

Parushu Vivo lenovo

Recording... Design Animations Slide Show Review View MultiType

```

1 // Method overriding.
2 class A {
3     int i, j;
4     A(int a, int b) {
5         i = a;
6         j = b;
7     }
8     // display i and j
9     void mogambo() {
10        System.out.println("i and j: " + i + " " + j);
11    }
12 }
13 // class B inherits from Class A
14
15 class B extends A {
16     int k;
17     B(int a, int b, int c) {
18         super(a, b);
19         k = c;
20     }
21     // display k - this overrides mogambo() in A
22     void mogambo() {
23         System.out.println("k: " + k);
24     }
25 }
26 public class Override {
27     public static void main(String args[]) {

```

Consider the following:

Top Class (super) void mogambo()

Derived class 1 void mogambo()

When mogambo() is invoked on an object of type B, the version of mogambo() defined within B is used. That is, the version of mogambo() inside B overrides the version

k: 73

Please move this window away from

Participants (44)

Find a participant

- Parushu (Host, me)
- lenovo (Co-host)
- 17D41A0430
- 17D41A0436
- 17D41A0464
- 17D41A04A7
- 17D41A04E0 VAMSHI
- 17D41A04E9
- 17D41A04J6\_CHARAN
- 17D41A04L0

6:50 PM 4/30/2021

Zoom Meeting

Parushu Vivo lenovo

Recording... Why Overridden Methods?(contd...)

Please move this window away from

Part of the key to successfully applying polymorphism is understanding that the superclasses and subclasses form a hierarchy which moves from lesser to greater specialization. Used correctly, the superclass provides all elements that a subclass can use directly. It also defines those methods that the derived class must implement on its own.

Participants (51)

Find a participant

19D45A0412	Ramya		
19D45A0417	Saketh		
19D45A0420			
19D45A0422			
19d45a0502			
B1	B.Anusha 17D41A0422		
CV	Chandana Varma		
H	Himaja		
S	S.sindhu		
V	vivek		

Invite Mute All

7:22 PM 4/30/2021