

# Nested If Statements

- The if-true-statement and if-false-statement of an if statement could be another if statement
- These are called nested *if statements*

```
if (a >= b)
    if (b >= c) min = c;
    else min = b;
else
    if (a >= c) min = c;
    else min = a;
```
- An else clause is matched to the last unmatched if (no matter what the indentation implies)

```
100 //package com.corejavaprograms;
101 package com.coreJavaPrograms;
102 class first {
103
104     public static void main(String[] args) {
105         int a=10;
106         int b=20;
107         if(a==10) {
108             if(b==20) {
109                 System.out.println("The Answer is correct");
110             }
111         }
112     }
113 }
114
115
116
117
118
119
120
121
```

Problems @ Javadoc Declaration Console

<terminated> first [Java Application] C:\Users\kumarneeraj\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86\_64\_17.0.5.v20221102-0933\jre\bin\javaw.exe (Jan 17, 2023, 2:59:07 PM - 2:59:12 PM) [pid: The Answer is correct

chater04 (1) [Compatibility Mode] - PowerPoint (Product Activation Failed) Neerai Kumar

File Edit View Navigate Code Refactor Build Run Tools VCS Window Help CoreJava - nestedifexample.java

CoreJava > src > com > corejava > nestedifexample > main

Project

- CoreJava C:\Users\kumarneeraj\IdeaProjects\CoreJava
  - .idea
  - out
  - src
    - com.corejava
      - nestedifexample
      - Main
  - External Libraries
  - Scratches and Consoles

nestedifexample.java

```
import java.util.Scanner;

no usages

class nestedifexample {
    no usages
    public static void main(String args[] ) {

        int age ;
        System.out.println("Enter your age");
        Scanner sc=new Scanner(System.in);
        age=sc.nextInt();
        if (age > 18) {
            System.out.println("You are Eligible to Vote! S Please vote");
        } else if (age <18) {
            System.out.println("You are Not Eligible to Vote..Please wait");
        } else if (age >=18) {

            System.out.println("You are Eligible but you have to wait for some months else please mention your
        } else {
            System.out.println("Good Luck Boys");
        }
    }
}
```

Run: first x nestedifexample x

Version Control Run TODO Problems Terminal Services Build

Build completed successfully in 1 sec, 367 ms (moments ago)

7:1 CRLF UTF-8 4 spaces

Slide 3 of 9 Accessibility: Unavailable

Notes Comments

10°C Sunny 5:26 PM 1/17/2023

# The Switch Statement

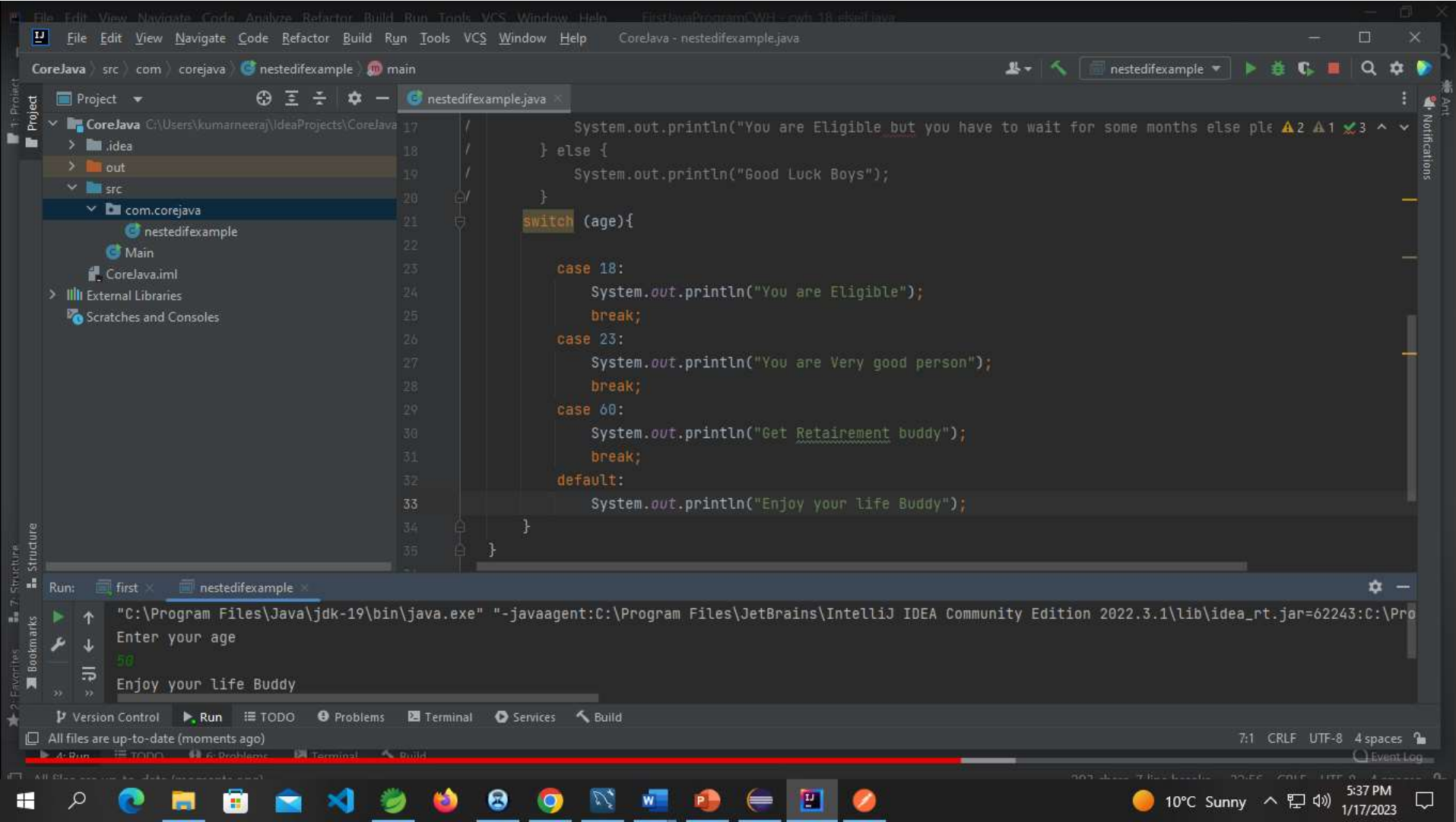
- The *switch statement* provides another means to decide which statement to execute next
- The switch statement evaluates an expression, then attempts to match the result to one of several possible *cases*
- Each case contains a value and a list of statements
- The flow of control transfers to statement list associated with the first value that matches

# The Switch Statement

- A switch statement can have an optional *default case* which has no associated value
- If the default case is present, control will transfer to it if no other case value matches
- The default case can be positioned anywhere in the switch, it is usually placed at the end
- If there is no default case, and no other value matches, control falls through to the next statement after the switch

# The Switch Statement

- Often a *break statement* is used as the last statement in each case's statement list
- A break statement causes control to transfer to the end of the switch statement
- If a break statement is not used, the flow of control will continue into the next case
- The expression of a switch statement must result in an integral data type, like an integer or character
- You cannot perform relational checks with a switch statement



# The For Statement

- The *for statement* has the following syntax:  
for ( *initialization* ; *condition* ; *increment* )  
    *statement* ;
- The *initialization* is executed once before the loop begins
- The *statement* is executed until the *condition* becomes false
- The *increment* portion is executed at the end of each iteration



# The For Statement

- The for statement is equivalent to the following while loop structure

```
initialization ;  
while ( condition )  
{  
    statement ;  
    increment ;  
}
```

- Like a while loop, the condition of a for loop is tested prior to executing the loop body
- Therefore, the body of a for loop can be executed zero or more times

# Do-while Statement

- A do-while statement checks the condition after executing the loop body
- The loop body of a do-while statement is executed at least once
- Do-while statements are suitable for writing loops that are executed at least once
- DoGrowth.java
- Babylonian.java (extra)