**Scope & Output Formatting, Switch Statement**

Reference: Gaddis 3.7, 3.9, 3.10

Topics: Scope, DecimalFormat, Switch statements

**Scope**

*Scope* defines when the variable is available to use

* A variable is one declared inside a method
* A variable is *in scope*
* What happens in the below code?

int value = name.length( );

if(value < 7)

int age = 18;

else if(value < 20)

int age = 25;

if(age < 20)

System.out.println(“young enough”);

else

System.out.println(“old enough”);

*What about in this code?*

int age = 0;

int value = name.length( );

if(value < 7)

age = 18;

else if(value < 20)

age = 25;

if(age < 20)

System.out.println(“young enough”);

else

System.out.println(“old enough”);

*What about this code?*

System.out.print(“Please enter your name”);

String name = scan.next();

String capname = name.toUpperCase();

if(capname.equals(“JIM”))

{

double score = 5.67;

int height = 6;

}

System.out.println(“Your score is ” + score);

**Decimal Formatting**

How do we make our double numbers not have tons of trailing zeros??

i.e. 40.00 instead of 40.00000000000001?

Need to import:

Algorithm:

Step 1: Declare a DecimalFormat object

Step 2: Instantiate the object to contain your format using the right *format pattern*

Step 3: Use it

Format Patterns:

Example:

You have an amount of money stored in variable “amount” and want it to be output in the format of $40.00.

//Declare DecimalFormat object

//Instantiate with format

//Output using the format we created

Example 2:

Format a number such that it has a comma in the right place up to the millionth place (i.e. 1,999,888 would have both commas):

DecimalFormat commas = new DecimalFormat(

Use the new format on variable “population” when outputting with System.out.print:

**Switch Statements -** What if I have some if/else-if statements on the same number?

*//Assume user has already entered two double numbers: fp1, fp2*

*//Print a menu, then prompt for the operation*System.out.println( "\n Operations are: \n\t M for multiplication \n\t D for division" );  
System.out.print( "Enter your selection: " );  
String operationS = scan.next();  
char operation = operationS.charAt( 0 );  
 *//perform the operation and print the result*if(operation == ‘M’)

{

double result = fp1 \* fp2;  
 System.out.println("The product is " + result );  
} else if(operation == ‘D’)

{  
 if ( fp2 == 0 )  
 System.out.println("Dividing by 0 is not allowed" );  
 else

{

double result = fp1/fp2;  
 System.out.println("The quotient is ” + fp1 / fp2);  
 }

} else  
 System.out.println( operation + " is not a valid option." );

SAME CODE AS SWITCH:

**switch** ( operation )  
{  
 **case** 'M':

double result = fp1 \* fp2;  
 System.out.println( "The product is " + result );  
 break;  
 **case** 'D':  
 if ( fp2 == 0 )  
 System.out.println( "Dividing by 0 is not allowed" );  
 else

{

double result = fp1/fp2;  
 System.out.println( "The quotient is ” + result );

}  
break;

**default**:  
 System.out.println( operation + " is not a valid option." );

}

So how did we make a switch statement?

**The switch statement** can replace the if-else if –else –if …

**switch** ( variable )

{

**case** constant1:

// do stuff here, can have as many statements as you want

**break**;

**case** constant2:

// do stuff here

**break**;

**case** constant3:

// do stuff here

**break**;

**default**:

// do stuff here

}

You can list multiple cases right after each other to allow multiple values for same code.

What if we want ‘m’ and ‘d’ to also be valid input?

**switch** ( operation )  
{  
 **case** 'M':

double result = fp1 \* fp2;  
 System.out.println( "The product is " + result );  
 break;  
 **case** 'D':  
 if ( fp2 == 0 )  
 System.out.println( "Dividing by 0 is not allowed" );  
 else

{

double result = fp1/fp2;  
 System.out.println( "The quotient is ” + result );

}  
break;

**default**:  
 System.out.println( operation + " is not a valid option." );  
}

What types of variables can you use in a switch statement?