Object Oriented Development with Java

(CT038-3-2-OODJ and Version VC1)

Data Types, Operators and Expressions

Copyright 2016 Asia Pacific Institute of Information Technology

Topic & Structure of the lesson

- **Overview of**
  - **Identifiers**
  - **Data types**
  - **Operators**
  - **Expressions**

Learning Outcomes

At the end of this topic, you should be able to:

- Define and differentiate the various data types in Java
- Use operators in Java programs and subsequently write expressions that make up a Java program

Identifiers/Variables

Variables

```
int temperature;  // The Fahrenheit
temperature
```

Think of variable like a container for a value :

temperature

```
temperature = 32;    // temperature contains the value 32
```

The above is an assignment statement and "=" is the assignment operator.

Identifiers/Variables

To declare > 1 variable :

int fahrTemp, centTemp;

int is the type name

Legal variable name must consists of a letter (upper- or lowercase) followed by any number (including zero) of letters & digits.

Illegal variable names : 4.7 !%--

Legal variable names :

temperature TEMP23 T $temp_1 T$$1

Identifiers/Variables

int Temp;

temp=3;

cause Java to give the error

Undefined variable; temp

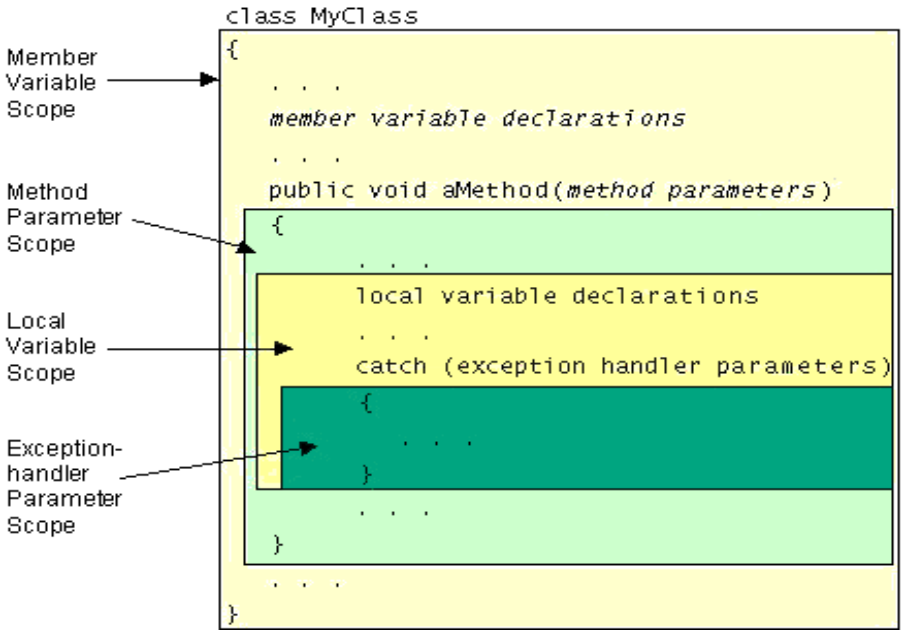
To declare a constant value :

final double PI = 3.14159;

Scope of Variables

A variable's scope is the block of code within which the variable is accessible and determines when the variable is created and destroyed. The location of the variable declaration within your program establishes its scope and places it into one of these 4 categories:

- Member variable
- Local variable
- Method parameter
- Exception-handler parameter



Keywords

- Keywords ~ words that may seems to be legal variable names but they are not because they are reserved by the language for special uses.
- List of Keywords in Java :

abstract	boolean	break	byte	case	cast
catch	char	class	const	continue	default
do	double	else	extends	false	final
finally	float	for	future	generic	goto
if	implements	import	inner	instanceof	int
interface	long	native	new	null	operator
outer	package	private	protected	public	rest
return	short	static	sure	switch	synchronized
this	throw	throws	transient	true	try
var	void	volatile	while		

Data types

All variables must have a data type and must be initialized.

Eg. int count=3;

or

```
int count;
count=3;
```

Data types

- Java language is rich in its data types.
- The variety of data types available allow the programmer to select the type appropriate to the needs of the application.
- Two major categories of data type :

  Primitive (Built in types)

  Reference / Object (Derived Types)

Data types – primitive data types

Type	Size/Format	Description
integers		
byte	8-bit two's complement	Byte-length integer
short	16-bit two's complement	Short integer
int	32-bit two's complement	Integer
long	64-bit two's complement	Long integer
Real numbers		
float	32-bit IEEE 754	Single-precision floating point
double	64-bit IEEE 754	Double-precision floating point
others		
char	16-bit Unicode character	A single character
boolean	true or false	A boolean value

Data types – primitive data types

Examples of **double** values in Java :

3.14159 7.12 9.0 0.5e001 -16.3e+002

The *e* is called the *e-notation* in Java;
e separates the number from the exponent.

2.829281×10^{8} <=> 2.829281e8

$2.13898121 \times 10^{-15}$ <=> 2. 13898121e-15

Data types – reference data types

- Reference types :
  - Arrays
  - Classes

  - The value of a reference type variable, is ***a reference to the actual value or set of values represented by the variable*.**

Data types – reference data types

- **Example – class references**

- **Classes are the templates for creating objects.**

```
class A
{
   variables
   constructors
   methods
}
```

A a = new A( ) a is a reference to an object of class A

Data types – reference data types

- **Example – Array references**

```
int myArray [ ] = new int [4];
```

Operators in Java

What is an operator?

- An operator takes one or more arguments (operands) and produces a new value
- An operator in Java can be
  - Unary: operates on a single operand
  - Binary: operates on 2 operands
  - Ternary: operates on 3 operands
- A Java operator can be further classified in accordance with the scheme as shown in the slide that follows

Operators in Java

Java supports a rich set of operators which can be classified into six categories:

1. Arithmetic operators
2. Relational operators
3. Logical operators
4. Increment/Decrement operators
5. Assignment operators

Operators in Java

Arithmetic Operators

– Java language supports the arithmetic operators as listed below for all integer and real numbers :

Operator	Use	Description
+	op1 + op2	Adds op1 and op2
-	op1 - op2	Subtracts op1 from op2
*	op1 * op2	Multiplies op1 by op2
/	op1 / op2	DIvide op1 by op2
%	op1 % op2	The remainder of dividing op1 by op2

Operators in Java

A · P · U
ASIA PACIFIC UNIVERSITY
OF TECHNOLOGY & INNOVATION

Relational

A relational operator compares 2 values and determines the relationship between them.

Operator	Use	Description
>	op1 > op2	Op1 is greater than op2
>=	op1 >= op2	Op1 is greater than or equal to op2
<	op1 < op2	Op1 is smaller than op2
<=	op1 <= op2	Op1 is smaller than or equal to op2
==	op1 == op2	Op1 is equal to op2
!=	op1 != op2	Op1 is not equal to op2

Operators in Java

Logical Operators

Relational operators are often used with logical operators to construct more complex decision-making expressions.

Operator	Use	Description
&&	op1 && op2	op1 and op2 are both true, *conditionally* evaluates op2
||	op1 || op2	Either op1 and op2 are true, *conditionally* evaluates op2
!	! op	op is false
&	op1 & op2	op1 and op2 are both true, *always* evaluates op2
|	op1 | op2	Either op1 and op2 are true, *always* evaluates op2
?:	expression ? op1 : op2	Shorthand for and ***if-else statement*** The ?: operator evaluates expression and returns op1 if it's true and op2 if it's false

Operators in Java

Increment/decrement Operators

++ increase value by 1

-- decrease value by 1

Eg.

i++ OR ++i

k- - OR - -k

Operators in Java

Assignment Operators

Assignment operators are used to assign one value to another. Listed below are the basic assignment operators (=) as well as the shortcut assignment operators.

Operator	Use	Description
=	op1 = op2	Op1 = op2
+ =	op1 += op2	op1 = op1 + op2
- =	op1 -= op2	op1 = op1 - op2
* =	op1 *= op2	op1 = op1 * op2
/ =	op1 /= op2	op1 = op1 / op2
%=	op1 %= op2	op1 = op1 % op2
&=	op1 &= op2	op1 = op1 & op2
| =	op1 |= op2	op1 = op1 | op2

Expressions

Definition : *An expression is* ***a series of variables, operators and method calls*** *(constructed according to the syntax of the language) that evaluates to a single value.*

- Expression perform the work of a Java program.
- Expressions are used to :
  - compute (eg. totalPrice = productCost + shippingCost)
  - assign values to variables (eg. count = 10)
  - to help control the execution flow of a program (eg while (count <10) count++;)

Expressions

Basically, there are 2 types of expressions :

Expression with operators

eg.

```
temperature = 98;
total = (count + 10)* 25 / 4;
count++;
```

Expressions

Method call expression

eg. keyboard.nextInt(),
Integer.parseInt()

A method(function) call evaluates to the return value of the method.

The return data type of a method expression call is the same as the data type of the return value of that method.

Follow Up Assignment

1. Write an expression that returns the solution for the general form of the quadratic equation as shown below

$$ax^2 + bx + c = 0$$

Quick Review Question

We are going to work together to write some more Java programs that we can test in the lab.

1. Write a Java program to display the lines:
   This is the first line.
   This is the second line.
2. Write a Java program to
   Assign the value 45.35 to the float variable price, 10 to the integer variable units and calculate and display the total value of price * units

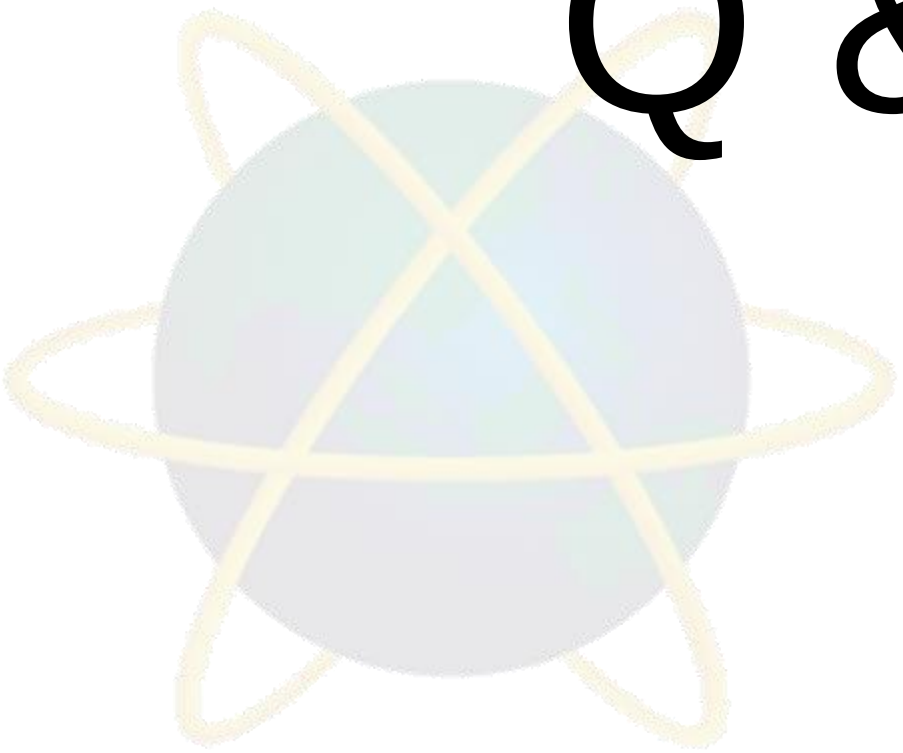
Summary of Main Teaching Points

- Identifiers
- Data types
- Operators
- Expressions

Question and Answer Session

A·P·U
ASIA PACIFIC UNIVERSITY
OF TECHNOLOGY & INNOVATION

Q & A



Next Session

- Conditional constructs
  - If… else construct
  - Nested if…else constructs
  - Switch…. Case
- Break and continue statements