Intro to Java Programming

Factsheets

Variables

Variable_Declaration	Comment	Note
int age = 21;	This declares an integer variable and initializes it to 21.	Recommended
int nextAge = age + 1;	The initial value of a variable can be an expression (as long as age has been previously declared.)	Recommended
String course= "Udacity";	The variable has type String and is assigned an initial value of "Udacity".	Recommended
score= 80;	ERROR: the type is required. This statement will not declare a variable. It is an assignment statement which assigns a new value to an existing variable.	NOT Recommended
int age= "42";	ERROR: You cannot initialize a number with a String. "42" is a String. See the quotation marks.	NOT Recommended
int age;	This declares an integer variable without initializing it. It is best to initialize variables when they are created: int age = 0; If you do not know what value you want yet	----

Naming Rule	Example
Names must consist of letters, numbers, an underscore, or a dollar sign only.	score_1
Don't use single letter variable name as you do in mathematics. While it is legal in Java, it is usually not a good idea because it can make programs harder to understand. (you will encounter a couple of exceptions later)	a
WARNING: Names are case sensitive. Note that by covention, variable names start with a lowercase letter	FinalGrade, finalGrade, and FINALGRADE are all different variables
ERROR Names cannot start with a number.	7up
ERROR. You cannot use a reserved word as a name.	int
ERROR: You cannot use special characters such as * or & in names.	m&m
ERROR: Names cannot contain spaces.	final grade

Number Types

Type	Range	Size
int (integer)	-2,147,483,648 to 2,147,483,647(~2.14 billion)	4 bytes
short (integer)	-32,768 to 32,767	2 bytes
long (integer)	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807	8 bytes
byte	-128 to 127	1 byte
double(double-precision floating point)	range of about + or - 10^308	8 bytes

Type	Range	Size
float(single-precision floating point)	range of about + or - 10^38 & about 7 significant decimal places	4 bytes
char	represents a Unicode character	2 bytes
boolean	has only 2 possible values: true or false	1 bit

Number Literals	Description	Examples
int	An integer has no fractional part and can be positive, negative, or 0.	5,-100,0
double	A number with fractional part	1.7, 1.0, 2.4E5, 3.47E-2
ERROR	Do not use a comma to separate thousands	1,000,000
ERROR	Do not use a fraction. Use a decimal instead.	3 1/4

Integer Arithmetic

Expression	Value (when n =2497)	Description
n/10	249	Notice that the answer is an integer with no decimal part.
n % 10	7	Always the last digit of n
n /100	24	Again, decimal part is discarded. Removes the last 2 digits.
n % 100	97	The last two digits.
n % 2	1	If n % 2 is 0 the number is even. Otherwise it is odd.

Math Functions

Method	Return Value
Math.sqrt (n)	Square root of n (if n is > or = to 0)
Math.pow(a,b)	a^b(if a = 0, b must be >0)
Math.sin(n)	Sine of n where n is in radians
Math.cos(n)	Cosine of n where n is in radians
Math.tan(n)	Tangent of n where n is in radians
Math.round(n)	closest integer to n as a long
Math.ceil(n)	smallest integer > or = to n as a double
Math.floor(n)	largest integer < or = to n as a double
Math.toRadians(n)	Converts n degrees to radians
Math.toDegrees(n)	Converts n radians to degrees
Math.abs(n)	Absolute value of n |n|
Math.max(a,b)	The larger of a and b
Math.min(a,b)	The smaller of a and b
Math.exp(n)	e^n
Math.log(n)	natural log of n
Math.log10(n)	Base 10 log of n

String Formatting

Code	In an Example	Type	What It Prints
d	"%4d"	Decimal integer	123
x	"%x"	Hexadecimal integer	7A

Code	In an Example	Type	What It Prints
o	"%o"	Octal integer	173
f	"%5.2f"	Fixed floating-point	12.30
e	"%e"	Exponential (very large or small) floating-point	1.23e+1
g	"%3.2g"	General (medium sized) floating-point	12.3
s	"%s"	String	Tax:
n	"%n" or "\n"	Line end	

Format Flags

Flag	In an Example	Meaning	What It Prints
-	"%-6d"	Align left	an integer that takes 6 spaces and starts in the first one
0	"%07.2f"	Show leading zeroes	0001.23
+	"%+7.2f"	Show a plus sign for positive numbers	+1.23
(	"%(6.2f"	Enclose negative numbers in parentheses	-1.23 would look like (1.23)
,	"%,10d"	Show decimal separators	12,300
^	"%^s"	convert letters to uppercase	"tax:" would print as "TAX:"

Strings

Example_Code_For_String_Methods	Result	Other info
String str = "Java "; str = str + "Programming"	str is assigned the value "Java Programming"	The + sign is used to concatenate Strings
String answer = "Total: " + 42;	answer is set to "Total: 42"	Because "Total: " is a string 42 is converted to a string and then the concatenation takes place
String name = "Sara T"; int len = name.length();	len is set to 6	The number of characters in a string. A space counts as a character
String city = "San Jose"; String sub = city.substring(1, 3);	sub is set to "an"	Takes the substring starting at position 1 and ending before position 3
String city = "San Jose"; String first = city.substring(0, 1);	first is set to "S"	Gets the first character. The substring has length 1
String city = "San Jose"; String sub = city.substring(4);"	sub is set to "Jose"	If you only supply one parameter, the substring consists of all characters from that position until the end of the String
String city = "San Jose"; String last = city.substring(city.length() - 1);	returns the string containing the last letter in the string ("e") and assigns it to last	str.substring(str.length() - 1) will always give you the last character as a String
String city = "San Jose"; int index = city.indexOf("Jose")	index is set to 4	returns the index where "Jose" starts
String city = "Santa Barbara"; int index = city.lastIndexOf("a")	index is set to 12	returns the index of the last "a" in the string
String cityWithTypo = "Son Jose"; String cityCorrected = cityWithTypo.replace("Son","San");	Changes all ocurrences of "Son" to "San" in cityWithTypo and put the result in cityCorrected	Will also worked the following ("So","Sa");

Example_Code_For_String_Methods	Result	Other info
String sentence = "Joseph is in San Jose"; int index = sentence.indexOf("Jose", 2)	index is set to 17	indexOf returns the index where "Jose" starts. When an index position is supplied as the second argument (2 in this case), search will begin AT that index

Common Loop Algorithms

Sum

```
total = 0
for each item
     total = total + input
```

Counting Matches

```
matches = 0
for each item
    if the item matches
        matches = matches + 1
```

Finding the Location of the First Match

```
found = false
position = 0
while it's not found, and there are more items
    if the item at position matches
        found = true
    else
        position = position + 1
if the item was found
    its location is position
```

Maximum

```
largest = the first item
for all the items except the first
    if the current item is larger than largest
        replace the value in largest with the current item
```

This page was last edited on 2016/07/18 09:57:05.

POPULAR NANODEGREE PROGRAMS +

STUDENT RESOURCES +

UDACITY +

INQUIRIES +