Unit 4 Lesson 2- Math functions

1. Basic math operations review

Syntax		*Example*	*Output*
% (modulo)	→	5 % 3	remainder 2
* (multiply)	→	2 * 2	equals 4
*= (multiply assign)	→	x*=2	x=2*x
+ (addition)	→	2+2	equals 4
++ (increment)	→	x++	x=x+1
+= (add assign)	→	x+=3	x=x+3
- (minus)	→	2-3	equals -1
- - (decrement)	→	x--	x=x-1
-= (subtract assign)	→	x-=3	x=x-3
/ (divide)	→	**3/5**	**equals 0**
	→	**3.0/5.0**	**equals 0.6**
/= (divide assign)	→	x/=3	x=x/3

Practice 1 (NO COMPUTER!)

In the following expressions determine the output. Indicate the data type of the output

A. 2 + 7 / 2	A. 5 (int)	A. 2.0 + 7 / (3 / 4)	A. 34/3 (double)
B. 3.5 (5 + 2.0)	B. 24.5 (float)	B. 5f * 8 / 10	B. 4f (int)
C. 5e02 - 02	C. 498 (float)	C. 35 / (5 / 4)	C. undefined
D. 5 * 3 / 2	D. 7 (int)	D. 4 - [6 - (2 - 7)]	D. 3 (int)
E. 2.0 * 7 / 2	E. 7 (float)	E. 5 / 6 * 2.0	E. 10/6 (double)
F. 2 * (3 / 4L)	F. 0 (int)	F. 5/ (6 * 2.0)	F. 5/12 (double)
G. (-9)%(-5)	G. 4 (int)	G. 51 % 50	G. 1 (int)
H. 20 / 7 + 20 % 7.0	H. 76/7 (double)	H. -10 % 2 + 1 / 2	H. 0.5 (float)
I. -9%(-2)/5	I. 0 (int)	I. 2.7 % 4	I. -1.3 (float)
J. 9 % 2. 5	J. 6.5 (float)	J. 2L + 5.7f % 1.2f	J. 2L+0.9f (float)
K. 1.5 % (3/4)	K. 0.75 (float)	K. 2 - 5/0.0	K. 2.0 (float)
L. 19 % 6	L.. 13 (int)	L. (-5) % (-1.5)	L. 0.5 (float)

2. Math methods used in calculations (selected functions only)

Function name	**Syntax**	**Use**
abs()	abs(-2)	Absolute value of any number

ceil()	float a=5.11;
	int b = ceil(a);
	ceil(9.04);
		Sets the value of a to 9
		
		Returns 10

floor()	float a=1.8;
	int b=floor(a);
		
		Returns 1

max()	max(1,2);
	max(1.1, 1.2);
	max(1,2,3);
		Returns 2
		Returns 1.2
		Returns 3
		Works with a single array in the same way

min()	min(1,2);
	min(1.1, 1.2);
	max(1,2,3);
		Returns 1
		Returns 1.1
		Returns 1
		Works with a single array in the same way

pow()	pow(4,2);
	pow(3,-5);
	pow(-3,5);
		Returns 16 (4*4)
		Returns (1 / 3*3*3*3*3) = 1/243
		Returns -3 *-3*-3*-3*-3 = -243

round()	round(9.2);	Returns 9 ; Rounds up or down according to rules
sqrt()	sqrt(16)	Returns 4 : Square root of a positive number only

random()	random(1,100)
	int w = int(random(1,100));
		Returns a random float number between 1 and 100. Must be cast to an integer if integer is needed

3. Other useful methods/syntax

A. **Convert a number to a String by concatenating an empty string to an int**

```
int numberZip = 19610;
String stringZipCode = numberZip + ""; // "19610"
```

B. **Convert a String to an int**

```
String stringZipCode = "19610";
int numberZipCode = Integer.parseInt(stringZipCode); // 19610
```

C. **Convert a String to a double**

```
String stringPrice = "19.99";
double numberPrice = Double.parseDouble(stringPrice); // 19.99
```

D. **Round a decimal number to the nearest whole number by casting:**

```
double unrounded = 12.56789;
println( (int) (unrounded + 0.5)); // 13
```

E. **Round a decimal number to the nearest tenth's place by casting:**

```
double unrounded = 12.56789;
println( (int) (unrounded * 10 + 0.5) / 10.0); // 12.6
```

F. **Round a decimal number to the nearest hundredth's place by casting:**

```
double unrounded = 12.56789;
println( (int) (unrounded * 100 + 0.5) / 100.0); // 12.57
```

G. Scientific notation

float x = 6.673e-11; **equals** float y = 0.00000000006673;

H. Increment/ decrement. Pre- and post- increment/decrement

Operator	Symbol	Example

assign addition	+=	a += 2;
		means: a = a + 2;

assign subtraction	-=	a -= 2;
		means: a = a - 2;

assign multiplication	*=	a *= 2;
		means: a = a * 2;

assign division	/=	a /= 2;
		means: a = a / 2;

assign mod	%=	a %= 2;
		means: a = a % 2;

post increment	++	a++; adds 1 to a, after operations involving a are completed eg. if (a++ < 2) -> a is incremented after the comparison is performed
pre increment	++	++a; adds 1 to a, before operations involving a are completed eg. if (++a < 2) -> a is incremented before the comparison is performed
post decrement	--	a--; same comment as a++
pre decrement	--	--a; same comment as ++a

4. (NO COMPUTER)Practice your knowledge by evaluating the following expressions
Subset A:

int i=4; int j=7	Values after execution
1. j += i;	1. i = 4; j = 11;
2. i -= j--;	2. i = -3; j = 6;
3. i *= --j;	3. i = 24; j = 6;
4. i *= j;	4. i = 28; j =7;
5. j %= --i;	5. i = 3; j = 1;
6. j /= ++i;	6. i = 5; j = 1.4;
7. j *= i-- % 3;	7. i = 0; j = 7;

Subset B:

1. abs(-5) - abs(-7)	1 -2	1. pow(16, 0.25)	1 2
2. abs(-1e-1) + abs(-2e-2)	2 0.12	2. pow(4, -2)	2 1/16
3. sqrt(0.0064)	3 0.08	3. round(1.49 + 0.1)	3 2
4. sqrt(pow(2.7, 2))	4 7.29	4. round(1.49) + 0.1	4 1.1
5. round(3.499)	5 3		