Module 4: Python Operators

Python supports the following types of operators:

- Arithmetic Operators
- Assignment Operators
- Relational Operators
- Boolean & Logical Operators
- Bitwise Operators
- Membership Operators
- Identity Operators

Arithmetic Operators:

Operators	Description
+	Addition
-	Subtraction
*	Multiplication
/	Division
//	Floor Division
%	Modulus
**	Exponent

// Floor Division- division of operands where result is quotient in which digits are removed after decimal point

P1

```
a = 7; b = 5;
print(a + b)
print(a-b)
print(a * b)
print(a / b)
print(a // b)
print(a % b)
print(a ** b)
```

P2

```
a = 5; b = 7;
print(a + b)
print(a-b)
print(a * b)
print(a / b)
print(a // b)
print(a % b)
print(a ** b)
```

Order of Operations:

- When more than one operator appears in expression Python follows **P E M D A S**.
- Operators with same precedence are evaluated left to right. (except Exponent operator which is right to left)

P1)	print( )	P2)	print( )
	print( )		print( )
	print( )		print( )
			print( )
P3)	print( )	P4)	print( )
	print( )		print( )

Assignment Operators:

Operator	Description
=	Assignment
+=	Compound Add
-=	Compound Subtraction
*=	Compound Multiply
/=	Compound Divide
//==	Compound Floor Division
%=	Compound Modulus
**=	Compound Exponent

	a = 6; b = 4;	Result
1	a += b; print(a)	
2	a -= b; print(a)	
3	a *= b; print(a)	
4	a /= b; print(a)	
5	a //= b; print(a)	
6	a %= b; print(a)	
7	a **= b; print(a)	

Relational Operators:

Operators	Description
==	Equal to
!=	Not Equal to
>	Greater than
>=	Greater than or equal to
<	Less than
<=	Less than or equal to

```
a , b = 10, 20
print(a == b)
print(a != b)
print(a < b)
print(a <= b)
print(a > b)
print(a >= b)
```

Boolean Operators:

Operators	Description
and	Boolean and
or	Boolean or
not	Boolean not

- X and Y ➔ if both X and Y are true then it returns True else False.
- X or Y ➔ if either X or Y are true then it returns True else False.
- not X ➔ if x is True it returns False else True

```
a = 10; b = 20; c = 30
print((a<b) and (c>b))
print((a<b) or (c>b))
print(not (a<b) )
```

Python Operator Summary:

Level	Operator	Description
18	()	Grouping
17	f()	Function call
16	[index:index]	Slicing
15	[]	Array Subscription
14	**	Exponential
13	~	Bitwise NOT
12	+ -	Unary plus / minus
11	* / %	Multiplication Division Modulo
10	+ -	Addition / Subtraction
9	<< >>	Bitwise Left Shift Bitwise Right Shift
8	&	Bitwise AND
7	^	Bitwise XOR
6	|	Bitwise OR
5	in , not in, is , is not <, <=, >, >= == !=	Membership Relational Equality Inequality
4	not	Boolean NOT
3	and	Boolean AND
2	or	Boolean OR
1	lambda	Lambda Expression

Module 4: Python Operators

Q1) What is the average value of the code that is executed below ?

```
grade1 = 80
grade2 = 90
average = (grade1 + grade2) / 2
```

Options:

a) 85 b) 85.0 c) 95 d) 95.0

Solution:

Q2) You develop a Python application for your company and you have the following code with line numbers:

```
01 def main(a, b, c, d):
02     value = a + b * c - d
03     return value
```

Q1) Which part of the expression will be evaluated first?

1) a + b 2) b * c 3) c – d

Q2) Which operation will be evaluated second?

1) Addition 2) subtraction

Q3) Which expression is equivalent to the expression in the function?

1) (a +b) * (c – d) 2) ((a+ (b * c)) - d) 3) a + ((b * c) – d)

Solution:

Q3) What is the value of the expression:

```
4+2**5//10
```

Options:

a) 3 b) 7

c) 77 d) 0

Solution:

Q4) Consider the expression given below.

The value of X is:

```
X = 2 + 9 * ( ( 3 * 12 ) - 8 ) / 10
```

Options:

a) 30.0 b) 30.8 c) 28.4 d) 27.2

Solution:

Q5) What is the result of the following expression:

```
(3 * (1+2) ** 2 - (2**2) * 3)
```

Options:

A. 3 B. 13

C. 15 D. 69

Solution:

Q6) You are writing a Python program to perform arithmetic operations. You create the following code:

```
a = 11
b = 4
```

What is the result of each arithmetic expression?

print(a/b) ; print(a//b); print(a%b)

Options:

A. 2 2 3 on same lines

B. 2 3 2.75 on separate lines

C. 2.75 2 3 on separate lines

D. 2 2 3 on separate lines

Solution:

Q7) Woodgrove Bank must generate a report that shows the average balance for all customers each day. The report must truncate the decimal portion of the balance. Which two code segments should you use? Each correct answer presents a complete solution. Choose two.

Options:

A. average_balance = total_deposits**number_of_customers

B. average_balance = total_deposits//number_of_customers

C. average_balance = int(total_deposits/number_of_customers)

D. average_balance = float(total_deposits//number_of_customers)

Solution: