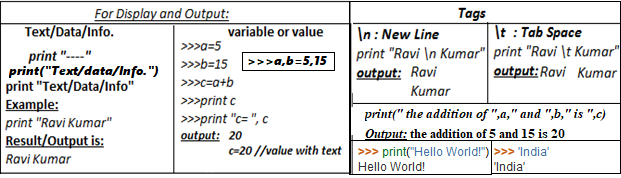
**Basic Structure and Operators**

****

**Using the Python interpreter** In our first code we are going to print “Hello World!” using the interpreter. To generate the output, type the following:  **>>> print("Hello World!") Hello World!**

**Variables and Assignment:** In algebra, variables represent numbers. The same is true in Python, except Python variables also can represent values other than numbers.

**Identifiers:** *While mathematicians are content with giving their variables one-letter names like x, programmers should use longer, more descriptive variable names. Names such as sum, height, and sub\_total are much better than the equally permissible s, h, and st. A variable’s name should be related to its purpose within the program. Good variable names make programs more readable by humans. Since programs often contain many variables, well-chosen variable names can render an otherwise obscure collection of symbols more understandable. Identifiers have the*

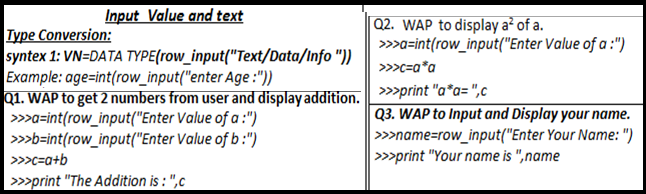
**Keywords and Identifiers:**The following identifiers are used as reserved words, or keywords of the language, and cannot be used as ordinary **identifiers.** They must be typed exactly as written here:

****

**Reading input from the Keyboard** input(“String to show”) // number = int(input("Enter an integer: "))

**The eval Function :** The input function produces a string from the user’s keyboard input. If we wish to treat that input as a number, we can use the int or float function to make the necessary conversion:

**x = float(input('Please enter a number'))**

****

**Operators:** Operators are the symbols which tells the Python interpreter to do some mathematical or logical operation. Few basic examples of mathematical operators are given below: **>>>** 2 + 3 5 , **>>>** 23 – 3 20

**EXPRESSIONS:** Generally while writing expressions we put spaces before and after every operator so that the code becomes clearer to: read, likea = 234 \* (45 - 56.0 / 34)

**IDENTIFIERS/VARIABLES** -Names given to any variable, function, class, union etc. Naming convention(rule) for writing identifier is as under: **1. Names of functions 2. Names of arrays 3. Names of variables 4. Names of classes**

i) First letter of identifier is always alphabet.

ii) Reserve word cannot be taken as identifier name.

iii) No special character in the name of identifier except under score sign ‘\_’.

Here are some examples of acceptable identifiers:

1. mohd
2. zara
3. abc
4. move\_na
5. a\_123
6. mynam
7. e50
8. \_temp
9. J
10. a23b9
11. retVal
12. **abc\_rr**

**SOME FACTS ABOUT IDENTIFIER :**

1. It is name given to program element.
2. Keywords cannot be used as Identifier.
3. Identifier name cannot start with a digit.
4. Key words cannot be used as a name.
5. Special Characters are not allowed
6. We can give any valid name to the identifier.
7. Upper case and lower case letters are distinct.
8. Global Identifier cannot be used as “Identifier”.
9. Identifier is the names are given by the programmer.
10. Only Alphabets, Digits and Underscores are permitted.
11. An identifier is used for any variable, function, data definition etc.
12. Other special characters are not allowed for naming a variable / identifier
13. PYTHON is case-sensitive so that Uppercase Letters and Lower Case letters are different
14. The name of identifier cannot begin with a digit. However, Underscore can be used as first character while declaring the identifier.
15. Only alphabetic characters, digits and underscore (\_) are permitted in PYTHON language for declaring identifier.

**ERRORS**

There are many types of error that are encountered during the program run. following are some of them:

*1.* ***COMPILER ERROR****.:* The errors encountered during the compilation process are called Compiler error. Compiler error are of two types • Syntax error. • Semantic error.

***SYNTAX ERROR****:* Syntax error is the one which appears when we commit any grammatical mistakes. These are the common error and can be easily corrected. These are produced when we translate the source code from high level language to machine language.

e.g **cot<<endl;** This line will produce a syntax error as there is a grammatical mistake in the word **cout**

***SEMANTIC ERROR:*** These errors appear when the statement written has no meaning.

e.g **a + b =c** ; this will result a semantically error as an expression should come on the right hand side of and assignment statement.

*2.* ***LINKER ERRORS.*** Errors appear during linking process e.g if the word **main** written as **mian** . The program will compile correctly but when link it the linking window will display errors instead of success.

*3.* ***RUN TIME ERROR:*** An abnormal program termination during execution is known as Run time Error.

e.g. If we are writing a statement X = ( A + B) /C ;

the above statement is grammatically correct and also produces correct result. But what happen if we gave value 0 to the variable c, this statement will attempt a division by 0 which will result in illegal program termination. Error will not be found until the program will be executed because of that it is termed as run time error.

3. Logical Error.: A logical error is simply an incorrect translation of either the problem statement or the algorithm. e.g : root1 = -b + sqrt(b \* b -4\*a\*c) / (2 \*a)

the above statement is syntactically correct but will not produce the correct answer because the division have a higher priority than the addition, so in the above statement division is performed first, then addition is performed but in actual practice to do addition performed then divide the resultant value by ( 2\* a).

**BASIC INPUT OUTPUT AND PROCESS**

1. WAP to add 2 numbers**[R=a+b]**
2. WAP to add 4 numbers**[R=a+b+c+d]**
3. WAP to multiply 3 numbers**[R=a\*b\*c]**
4. WAP to find average 5 numbers**[R=(a+b+c+d+e)/5]**
5. WAP to display age after 15 years.**[nage =age+15]**
6. WAP to find the area of squire .**[A=a\*a]**
7. WAP to find the area of rectangle **[A=a\*b]**
8. WAP to display a3 numbers **[R=a\*a\*a]**
9. WAP to find the area of circle **[A=3.14\*r\*r]**
10. WAP to find the volume of sphere**.[v=4/3\*3.14\*r3]**
11. WAP to find the perimeter of rectangle**[A=2\*(l+ b) ]**
12. WAP to find the circumference of circle **[C=2\*3.14\*r]**
13. WAP to calculate simple interest. **[SI=PRT/100]**
14. WAP to find gross salary **[GS=BASIC+DA-PF+HRA]**
15. WAP to calculate sum of 5 subjects & find percentage**[TOT=s1+s2+s3+s4+s5,Per=TOT/5]**
16. WAP which accept temperature in Fahrenheit and print it in centigrade**[c=5/9\*(T-32)]**
17. WAP which accept temperature in centigrade and print it in Fahrenheit**[F=(1.8\*T)+32]**
18. WAP to input cost and display cost after increasing 25% **[cost+(cost\*25)/100]**
19. WAP to input Hours, Minutes and Seconds and display in seconds.**[TS=H\*60\*60+M\*60+S]**
20. WAP to take principal, rate and time and display C.I. (Compound Interest) CI=p\*(1+R/100)^T

**Q1. Classify each of the following as either a legal or illegal Python identifier:**

1. fred #Keywords
2. #if # Invalid
3. #2x #Invalid
4. -4 #Constant
5. sum\_total #identifier
6. sumTotal #identifier
7. sum-total #Expression
8. #sum total #Invalid
9. Sumtotal #identifier
10. While #identifier
11. x2 #identifier
12. Private #Keywords
13. public #Keywords
14. #$16 #Invalid
15. xTwo #identifier
16. \_static #identifier
17. \_4 #identifier
18. \_\_\_ #identifier
19. #10% #Invalid
20. a27834 #identifier
21. #wilma’s #Invalid

**Q2. If x = 2 Indicate what each of the following Python statements would print.**

|  |  |
| --- | --- |
| print("x") | 1. x 2. x 3. 2 4. x + 1 5. #Invalid 6. 3 |
| print(’x’) |
| print(x) |
| print("x + 1") |
| print(’x’ + 1) |
| print(x + 1) |

**Q3. Find Output: if *i1 = 2,i2 = 5,i3 = -3,d1 = 2.0,d2 = 5.0,d3 = -0.5;***

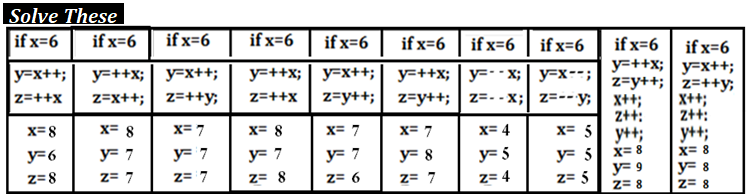
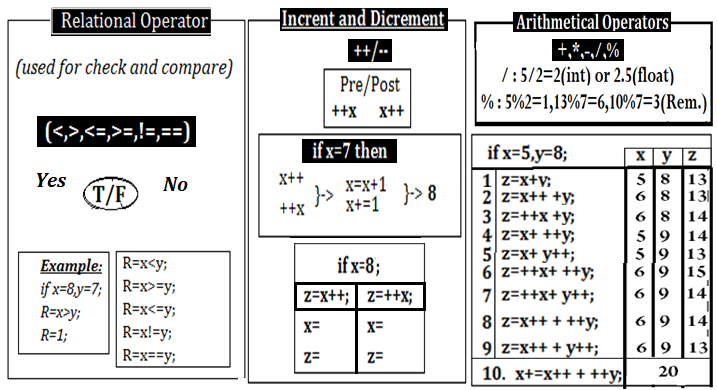
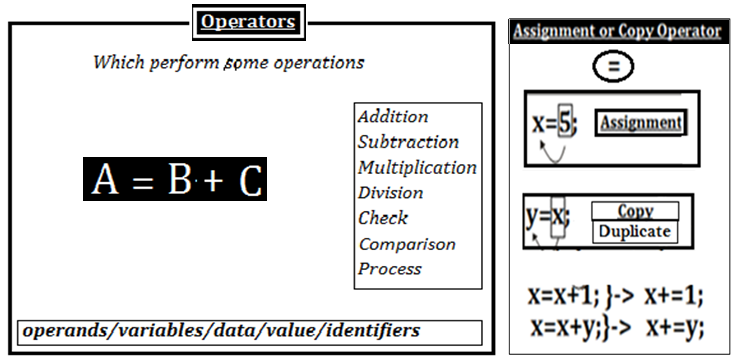
|  |  |  |
| --- | --- | --- |
| 1. i1 + i2 2. i1 / i2 3. i1 // i2 4. i2 / i1 5. i2 // i1 6. i1 \* i3 7. d1 + d2 8. d1 / d2 9. d2 / d1 10. d3 \* d1 11. d1 + i2 12. i1 / d2 13. d2 / i1 14. i2 / d1 15. i1/i2\*d1 16. d1\*i1/i2 17. d1/d2\*i1 18. i1\*d1/d2 19. i2/i1\*d1 20. d1\*i2/i1 21. d2/d1\*i1 22. i1\*d2/d1 | 1. print(i1 + i2) 2. print(i1 / i2) 3. print(i1 // i2) 4. print(i2 / i1) 5. print(i2 // i1) 6. print(i1 \* i3) 7. print(d1 + d2) 8. print(d1 / d2) 9. print(d2 / d1) 10. print(d3 \* d1) 11. print(d1 + i2) 12. print(i1 / d2) 13. print(d2 / i1) 14. print(i2 / d1) 15. print(i1/i2\*d1) 16. print(d1\*i1/i2) 17. print(d1/d2\*i1) 18. print(i1\*d1/d2) 19. print(i2/i1\*d1) 20. print(d1\*i2/i1) 21. print(d2/d1\*i1) 22. print(i1\*d2/d1) | 1. 7 2. 0.4 3. 0 4. 2.5 5. 2 6. -6 7. 7.0 8. 0.4 9. 2.5 10. -1.0 11. 7.0 12. 0.4 13. 2.5 14. 2.5 15. 0.8 16. 0.8 17. 0.8 18. 0.8 19. 5.0 20. 5.0 21. 5.0 22. 5.0 |

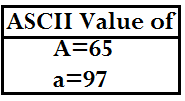
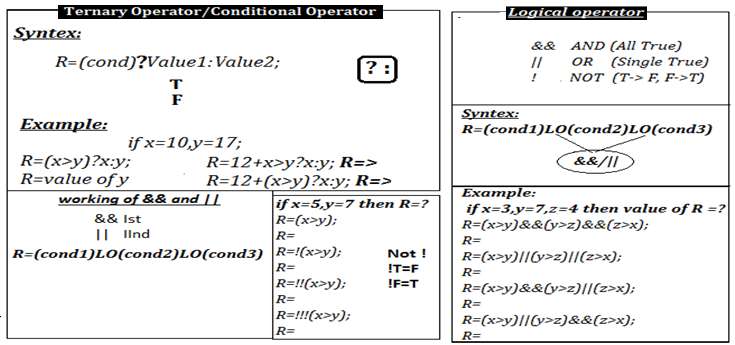
**Q4. Find Output: if i2 = 5,i3 = -3,1 = 2.0,d2 = 5.0,d3 = -0.5**

|  |  |  |
| --- | --- | --- |
| 1. i1 + (i2 \* i3) 2. i1 \* (i2 + i3) 3. i1 / (i2 + i3) 4. i1 // (i2 + i3) 5. i1 / i2 + i3 6. i1 // i2 + i3 7. 3 + 4 + 5 / 3 8. 3 + 4 + 5 // 3 9. (3 + 4 + 5) / 3 10. (3 + 4 + 5) // 3 11. d1 + (d2 \* d3) 12. d1 + d2 \* d3 13. d1 / d2 - d3 14. d1 / (d2 - d3) 15. d1 + d2 + d3 / 3 16. (d1 + d2 + d3) / 3 17. d1 + d2 + (d3 / 3) 18. 3 \* (d1 + d2) \* (d1 - d3) | 1. print(i1 + (i2 \* i3)) 2. print(i1 \* (i2 + i3)) 3. print(i1 / (i2 + i3)) 4. print(i1 // (i2 + i3)) 5. print(i1 / i2 + i3)) 6. print(i1 // i2 + i3) 7. print(3 + 4 + 5 / 3) 8. print(3 + 4 + 5 // 3) 9. print( (3 + 4 + 5) / 3) 10. print( (3 + 4 + 5) // 3) 11. print(d1 + (d2 \* d3)) 12. print(d1 + d2 \* d3) 13. print(d1 / d2 - d3) 14. print(d1 / (d2 - d3)) 15. print(d1 + d2 + d3 / 3) 16. print( (d1 + d2 + d3) / 3) 17. print(d1 + d2 + (d3 / 3)) 18. print(3 \* (d1 + d2) \* (d1 - d3)) | 1. -13 2. 4 3. 1.0 4. 1 5. -2.6 6. -3 7. 8.666666666666666 8. 8 9. 4.0 10. 4 11. -0.5 12. -0.5 13. 0.9 14. 0.36363636363636365 15. 6.833333333333333 16. 2.1666666666666665 17. 6.833333333333333 18. 52.5 |

**Q5. Write the shortest way to express each of the following statements.**

|  |  |
| --- | --- |
| 1. x = x + 1 | X+=1  x/=2  x-=1  x+=y  x-=y+7  x\*=2 |
| 1. x = x / 2 |
| 1. x = x - 1 |
| 1. x = x + y |
| 1. x = x - (y + 7) |
| 1. x = 2\*x |



******

***Increment and Decrement if x=13, r=11,p=10, q=8then find the value of x, r, p, q after execution?***

1. r+=p\*++x + ++q; x=14, r=160,p= 10, q=9
2. q-=x++ + ++x; x=15, r=11,p= 10, q=-20
3. r=q++ + p--; x=13, r=18,p= 9, q=9
4. p+=++x\*q++ + ++p; x=14, r=11,p= 134, q=9
5. r=p+ ++q+ --q +x; x=13, r=40,p= 10, q=8
6. r=(p++\*q)\*x++ -r++; x=14, r=1029,p= 11, q=8
7. q=(++p\*x++)+r +q++; x=14, r=11,p= 11, q=162
8. x=(++x)+r; x=25, r=11,p= 10, q=8
9. x+=(p+’F’+’a’+p++); x=200, r=11,p= 11, q=8
10. r-=(‘a’+’J’+++x); x=14, r=-174,p= 10, q=8

***If x=11,r=3; then find the value of r, x=?***

1. r=10\*++x + 1; r=121, x=12
2. r+=++x + x++; r=27, x=13
3. r=r\*x++ + x; r=45, x=12
4. r=++x\*2; r=24, x=12
5. r=x++\*2; r=22, x=12
6. r=(x++)\*x++ -r; r=129, x=13
7. r=(2\*x++)+r; r=25, x=12
8. r=(2\*++x)+r; r=27, x=12
9. r=(‘A’+’a’+x); r=173, x=11
10. r=('A'+'b'+x++) ; r=174, x=12

***if a=7,p=10,q=7 find the value of a, p, q=?***

1. p+=++a + --a +q\*a; a=7, p=74, q=7
2. p\*=q+ ++a + ++p ; a=8, p=286, q=7
3. p+= q++ + ++a- --q; a=8, p=18, q=7
4. p=p\*q/a-p; a=7, p=0, q=7

**Ternary Operator/Conditional Operator Find the value of k=? After execution the statement.**

1. k=10+50>1500?20:30; k= 30
2. k=200<100?21:20; k= 20
3. k=200+100<=100?101:11; k= 11
4. k=300+400>=500?112:222; k= 112
5. k=10>=10?20\*2:30+2; k= 40
6. k=2\*(1200>100?101:20); k= 202
7. k=1000+500>1500?25:20; k= 20
8. k=100+500>100?21:10; k= 21
9. k=200+200<100?10:11; k= 11
10. k=500+1000>500?12:22; k= 12
11. k=3+4-7\*(200+100<=100?101:11); k= -70
12. k=3\*4/3\*7+400>=500?112:222; k= 222

**Logical Operator if the a =5, b =6 , c= 7 then find the value of Exp=? After execution**

Exp =!!( a>b) || (a<c)&&(a>b); Exp = 0

Exp = (a<b) || !( b>c); Exp = 1

Exp = !(b>c)\*10; Exp = 10

Exp = (a>c) || (b<c) &&!!!(a>c); Exp = 1

Exp =!!(a>c)&&!!(b<c)&&!!(a>b); Exp =

Exp = ( a>b) && (a<c); Exp = 0

Exp = (a<b) || ( b>c); Exp = 1

Exp = !(b>c); Exp = 1

Exp = (a<c) && (b<c) || !(a>c); Exp = 1