PYTHON – WORKSHEET 1

1. Which of the following operators is used to calculate remainder in a division?

A) #

B) &

C) %

D) $

Answer: (C) %

2. In python 2//3 is equal to?

A) 0.666 B) 0 C) 1 D) 0.67

Answer: (B) 0

3. In python, 6<<2 is equal to?

A) 36 B) 10 C) 24 D) 45

Answer: (C) 24

4. In python, 6&2 will give which of the following as output?

A) 2 B) True C) False D) 0

Answer: (A) 2

5. In python, 6|2 will give which of the following as output?

A) 2 B) 4 C) 0 D) 6

Answer: (D) 6

6. What does the finally keyword denotes in python?

A) It is used to mark the end of the code

B) It encloses the lines of code which will be executed if any error occurs while executing the lines of code in the try block.

C) the finally block will be executed no matter if the try block raises an error or not.

D) None of the above

Answer: (B) It encloses the lines of code which will be executed if any error occurs while executing the lines of code in the try block.

7. What does raise keyword is used for in python?

A) It is used to raise an exception.

B) It is used to define lambda function

C) it's not a keyword in python.

D) None of the above

Answer: (A) It is used to raise an exception.

8. Which of the following is a common use case of yield keyword in python?

A) in defining an iterator

B) while defining a lambda function

C) in defining a generator

D) in for loop.

Answer: (A) in defining an iterator

9. Which of the following are the valid variable names?

A) _abc

B) 1abc

C) abc2

D) None of the above

Answer: (A) _abc & (C) abc2

10. Which of the following are the keywords in python?

A) yield

B) raise

C) look-in

D) all of the above

Answer: (A) yield & (B) raise

In [13]:
```
11
```

Out[13]:
```
11
```

In [2]:
```
def factorial(x):
    if x == 1:
        return 1
    else:
        return (x * factorial(x-1))

num = 7
result = factorial(num)
print("The factorial of", num, "is", result)
```

```
The factorial of 7 is 5040
```

In [14]:
```
12
```

Out[14]:
```
12
```

In [15]:
```
num = 31

flag = False

if num == 1:
    print(num, "is not a prime number")
elif num > 1:
    for i in range(2, num):
        if (num % i) == 0:
            flag = True
            break

    if flag:
        print(num, "is not a prime number")
    else:
        print(num, "is a prime number")
```

```
31 is a prime number
```

In [5]:
```
13
```

Out[5]:
```
13
```

In [12]:
```
my_str = 'aIbohPhoBiA'

my_str = my_str.casefold()

rev_str = reversed(my_str)

if list(my_str) == list(rev_str):
   print("The string is a palindrome.")
else:
   print("The string is not a palindrome.")
```

```
The string is a palindrome.
```

In [6]:
```
14
```

Out[6]:
```
14
```

In [7]:
```
def pythagoras(opposite_side,adjacent_side,hypotenuse):
        if opposite_side == str("x"):
            return ("Opposite = " + str(((hypotenuse**2) - (adjacent_side**2))**0.5))
        elif adjacent_side == str("x"):
            return ("Adjacent = " + str(((hypotenuse**2) - (opposite_side**2))**0.5))
        elif hypotenuse == str("x"):
            return ("Hypotenuse = " + str(((opposite_side**2) + (adjacent_side**2))**0.5))
        else:
            return "Actual answer!"

print(pythagoras(3,4,'x'))
print(pythagoras(3,'x',5))
print(pythagoras('x',4,5))
print(pythagoras(3,4,5))
```

```
Hypotenuse = 5.0
Adjacent = 4.0
Opposite = 3.0
Actual answer!
```

In [8]:
```
15
```

Out[8]:
```
15
```

In [11]:
```
test_str = "Go somewhere"

all_freq = {}

for i in test_str:
    if i in all_freq:
        all_freq[i] += 1
    else:
        all_freq[i] = 1


print("Count of all characters in GeeksforGeeks is :\n "
      + str(all_freq))
```

```
Count of all characters in GeeksforGeeks is :
 {'G': 1, 'o': 2, ' ': 1, 's': 1, 'm': 1, 'e': 3, 'w': 1, 'h': 1, 'r': 1}
```

In [ ]:
```

```