**PYTHON**(worksheet-4)

1. (B)
2. (B)
3. (B)
4. (D)
5. (C)
6. (B)
7. (C)
8. (A)
9. (A, D)
10. (A, B, D)
11. **Interpreter:** it translates only one statement of the program at a time into machine code. It translates program continuously till the first error is found. Hence, debugging becomes easy. Overall time taken by interpreter to execute the process is very slow. Interpreter does not generate an intermediary code. hence, it is highly efficient in terms of its memory.

**Compiler:** it translates whole of the program into machine code at once. It generates the error only after the complete executing the whole program. Hence, debugging becomes difficult as compared to interpreter. Due to it executes whole program at once so overall time taken by compiler to execute the process is much faster. It always generates an intermediary object code. So, it needs more memory.

So, interpreter is used in python language.

1. PYTHONPATH is used to assist in import module lookup. So, when we import modules, PYTHONPATH is also looked into to check which directories might contain the imported module.
2. By using strip() function we can remove all the leading and trailing whitespaces in a string in python.

**Example:** string = ' python is a very interactive language '

string.strip()

**output:** ‘python is a very interactive language’

1. #python program to represent a user entered number in expanded form

num= int(input("enter a number:"))

def expands(num):

digits=str(num)

output = []

for i, digit in enumerate(digits):

output.append("(" + digit + "x10^" + str(len(digits)-i-1) + ")")

return " + ".join(output)

expands(num)

1. #check whether the number is an Armstrong number or not

num = int(input("enter a number:"))

order = len(str(num))

sum = 0

temp = num

while temp > 0:

digit = temp % 10

sum += digit \*\* order

temp //= 10

if num == sum:

print(num,"is an Armstrong number")

else:

print(num,"is not an Armstrong number")