**Python**

**Assignment – 1**

Name: Mukesh Choudhary

Email: [2018bcamaismukesh6356@poornima.edu.in](mailto:2018bcamaismukesh6356@poornima.edu.in)

**Class 1:-**

Q1. What is JPython & CPython?

Answer:

* **CPython**: CPython is the implementation of the language called “Python” in C. not only Cpython, some more are implemented like IronPython and Jython (Python implemented in Java).
* **JPython**: Python is an interpreted programming language. Hence, Python programmers need interpreters to convert Python code into machine code. Where as Cython is a compiled programming language. The Cython programs can be executed directly by the CPU of the underlying computer without using any interpreter.

Q2. Basic difference between Python2 & python3

Answer:

|  |  |  |
| --- | --- | --- |
| **Basis of comparison** | **Python 3** | **Python 2** |
| Release Date | 2008 | 2000 |
| Function print | print ("hello") | print "hello" |
| Division of Integers | Whenever two integers are divided, you get a float value | When two integers are divided, you always provide integer value. |
| Unicode | In Python 3, default storing of strings is Unicode. | To store Unicode string value, you require to define them with "u". |
| Syntax | The syntax is simpler and easily understandable. | The syntax of Python 2 was comparatively difficult to understand. |
| Rules of ordering Comparisons | In this version, Rules of ordering comparisons have been simplified. | Rules of ordering comparison are very complex. |
| Iteration | The new Range() function introduced to perform iterations. | In Python 2, the xrange() is used for iterations. |
| Exceptions | It should be enclosed in parenthesis. | It should be enclosed in notations. |
| Leak of variables | The value of variables never changes. | The value of the global variable will change while using it inside for-loop. |
| Backward compatibility | Not difficult to port python 2 to python 3 but it is never reliable. | Python version 3 is not backwardly compatible with Python 2. |
| Library | Many recent developers are creating libraries which you can only use with Python 3. | Many older libraries created for Python 2 is not forward-compatible. |

Q3. Difference between ASCII & Unicode

Answer:

|  |  |
| --- | --- |
| **ASCII** | **Unicode** |
| ASCII character defines 128 characters | UNICODE defines 221 characters |
| ASCII is not standardized | Unicode is standardized |
| ASCII is stored as 8-bit byte | Unicode is stored in byte sequence such as UTF-32 and UTF-8 |
| ASCII uses numbers to represent text. Digits (1,2,3, etc.), letters (a, b, c, etc.) and symbols (!) are called characters | Unicode contains representations for letters in languages such as English, Greek, Arabic etc., mathematical symbols, emoji and many more |

**Class 2:-**

Q1. What should be the output? ( 3 + 4 \*\* 6 - 9 \* 10 / 2 )

Answer:

1054

( 3+ 1096 – 9\*10/2)

( 3+ 1096 – 9\*5)

( 3+ 1096 – 45)

(1099 – 45)

1054

Q2. Let say I have, some string "hello this side regex" find out the count of the total vowels - ['a','e','i','o','u']

Answer:

str = "hello this side regex"

s = str.lower()

vowels=0

for i in s:

if(i=='a' or i=='e' or i=='o' or i=='i' or i=='u'):

vowels=vowels+1

print (vowels)

Q3. Find out the area of triangle

- 1/2 \* b \* h (formula of area)

- You have to take value from user about the base, & the height

Answer:

print("Enter the base of the triangle: ")

base = int(input())

print("Enter the height of the triangle: ")

height = int(input())

area = 0.5\*base\*height

print("Area: ")

print (area)

Q4. Print the calendar on the terminal. If you give the year.

- Allow the user to input the year.

- Then should that calendar of that year

Answer:

import calendar

print("Enter year: ")

y = int(input())

print(calendar.calendar(y))

**Class 3:-**

Q1. Find the Armstrong Number between the two numbers which are input by user.

Answer:

lower = int(input("Enter the lower limit: "))

upper = int(input("Enter the upper limit: "))

for num in range(lower,upper + 1):

sum = 0

temp = num

while temp > 0:

digit = temp % 10

sum += digit \*\* 3

temp //= 10

if num == sum:

print(num)

Q2. Remove the punctuation like [“@!#$%&\*()”] from the string and output the string without them

Answer:

punc = '''!()-[]{};:'"\,<>[./?@#$%^&\*\_~](mailto:./?@#$%^&*_~)'''

my\_str = str(input("Enter the string: "))

no\_punc = ""

for char in my\_str:

if char not in punc:

no\_punc = no\_punc + char

print(no\_punc)

Q3. Sort the list of words in Alphabetical order.

Answer:

my\_str = ['Apple', 'banana', 'cat', 'REGEX','apple']

my\_str.sort()

print(my\_str)