UE18CS101:
INTRODUCTION TO COMPUTING USING

Department of Computer Science and Engineering
PES UNIVERSITY

PES UNIVERSITY

Lecture 12

Concept of Library, Problem solving using different libraries

- The “Python library” contains several different kinds of components.

- It contains data types that would normally be considered part of the “core” of a language, such as numbers and lists.
- For these types, the Python language core defines the form of literals and places some constraints on their semantics, but does not fully define the semantics. (On the other hand, the language core does define syntactic properties like the spelling and priorities of operators.)

- Python contains a large *library* of standard functions which can be used for common programming tasks (You can also create your own).
- The functions in the library are contained in separate *modules*, similar to the ones you have been writing and saving in the editor so far. In order to use a particular module, you must explicitly *import* it. This gives you access to the functions it contains.

Math Module

- >>>import math
- >>> dir(math)
- ['__doc__', '__file__', '__loader__', '__name__', '__package__', '__spec__', 'acos', 'acosh', 'asin', 'asinh', 'atan', 'atan2', 'atanh', 'ceil', 'copysign', 'cos', 'cosh', 'degrees', 'e', 'erf', 'erfc', 'exp', 'expm1', 'fabs', 'factorial', 'floor', 'fmod', 'frexp', 'fsum', 'gamma', 'gcd', 'hypot', 'inf', 'isclose', 'isfinite', 'isinf', 'isnan', 'ldexp', 'lgamma', 'log', 'log10', 'log1p', 'log2', 'modf', 'nan', 'pi', 'pow', 'radians', 'sin', 'sinh', 'sqrt', 'tan', 'tanh', 'tau', 'trunc']

Random Module

- >>>import random
- >>>dir(random)
- ['BPF', 'LOG4', 'NV_MAGICCONST', 'RECIP_BPF', 'Random', 'SG_MAGICCONST', 'SystemRandom', 'TWOPI', '_BuiltinMethodType', '_MethodType', '_Sequence', '_Set', '__all__', '__builtins__', '__cached__', '__doc__', '__file__', '__loader__', '__name__', '__package__', '__spec__', '_acos', '_bisect', '_ceil', '_cos', '_e', '_exp', '_inst', '_itertools', '_log', '_pi', '_random', '_sha512', '_sin', '_sqrt', '_test', '_test_generator', '_urandom', '_warn', 'betavariate', 'choice', 'choices', 'expovariate', 'gammavariate', 'gauss', 'getrandbits', 'getstate', 'lognormvariate', 'normalvariate', 'paretovariate', 'randint', 'random', 'randrange', 'sample', 'seed', 'setstate', 'shuffle', 'triangular', 'uniform', 'vonmisesvariate', 'weibullvariate']

Library

```
import math
math.sqrt(25)
math.sin(value in radian)
Math.radians(andle in degree)

import random
random.randrange(0,11)
random.randrange(0,1001,5)
```

Math Library

Syntax: import math
math.floor(x)
Parameter: x-numeric expression.

Returns: largest integer not greater than x.

```
import math
print (math.e)
print(math.pi)
```

```
import math
math.ceil(x)
```

Parameter: x:This is a numeric expression.

Returns: Smallest integer not less than x

Math Library

import math
math.exp(x)

Returns e^x

import math
math.fabs(x)

Returns

Absolute value

math.factorial(x)
Parameters : x : The number whose factorial has to be computed.
Return value : Returns the factorial of desired number.
Exceptions : Raises Value error if number is **negative or non-integral**

math.gcd(x, y)

truncate which behaves as a ceiling function for negative number and floor function for positive number.
In case of positive number

```
import math
print (math.trunc(3.5)) # work as floor
print (math.trunc(-3.5))  # ceil
```

import random

choice(seq)

Choose a random element from a non-empty sequence.

randint(a, b)

Return random integer in range [a, b], including both end points.

randrange(start, stop=None, step=1)

| Choose a random item from range(start, stop[, step]).
| This fixes the problem with randint() which includes the endpoint; in Python this is usually not what you want.

sample(population, k)

Chooses k unique random elements from a population sequence or set.

seed(a)

| Initialize internal state from hashable object.
| None or no argument seeds from current time or from an operating
| system specific randomness source if available.

import random

shuffle(x)

| Shuffle list x in place, and return None.

uniform(a, b)

| Get a random number in the range [a, b) or [a, b] depending on rounding.