National Institute of Technology Karnataka, Surathkal

Indian Society For Technical Education

NITK Chapter

Python SMP session – 5b Modules and Packages

Modules:

Modules help increase the organization of code, grouping related functions in modules, increases readability and understandability of code. Essentially,a module is a file containing definitions of Python functions, classes, variables. A module maybe user-defined or builtin. I have defined a python file speaker.py in the same directory as of the main file.

```
import speaker

speaker.welcome("ISTE")#Welcome,ISTE
```

Now, how does import know where the file speaker.py is located? What happens if speaker.py is not located in the same directory as bot.py (the main file which is executed)? To answer this question, a special variable sys.path is used to store a list of strings that specifies the search path for modules. So whenever import is called, Python first checks the current directory for the module that is getting imported then it checks every path in the sys.path the module getting imported.

```
import sys

sys.path.append(path_to_speaker)

import speaker

speaker.welcome("ISTE")#Welcome,ISTE
```

To import specific functions from modules, the from keyword can be used. To import every function, variable, class from a module, use the * operator, for example:

```
From speaker import welcome
welcome('ISTE')#Welcome,ISTE

from speaker import*
welcome('ISTE')#Welcome,ISTE
goodbye('ISTE')#Goodbye,ISTE
```

Packages:

Multiple modules working together can all be defined under an umbrella term called packages. A package is basically a hierarchical file directory structure that contains many .py files which may be independent modules. For example, assume there is a directory named Phone which

consists of three files - call.py, sms.py,init.py. initpy tells python that the current folder is a python package, You can leave

init.py empty.

```
#phone/sms.py
def send_sms(msg):
      print("SMS sent with message=",msg)
def receive_sms(msg):
      print("SMS received with message=",msg)

#phone/call.py
def make_call(number):
      print("Calling ",number)
def receive_call(number):
      print("Call from ",number)
```

We can import modules from packages using import statement.

```
import phone.call,phone.sms
phone.call.make_call("123456789")
phone.sms.receive_sms("How was your ISTE SMP")#Calling #123456789
#SMS received with message= How was your ISTE SMP?
```

Note that is necessary to full path of module
You can use from phone import * to import all modules from package.
but before doing that you to specify which modules you want directly to be imported,you can do this by assigning all variable to names of modules you want to import.

```
#phone/__init__.py
__all__=["call","sms"]

from  phone import *
call.make_call("9035230091")#Calling 9035230091
sms.receive_sms("How was your ISTE SMP?")#SMS received
#with message= How was your ISTE SMP?
```

External Packages:

Python has a variety of packages developed by other developers and made available on the Python Package Index, also called PyPI. pip can be used to install python libraries like, numpy, pandas etc.. To install any package from PyPI, one must use the pip command from the terminal. To install the latest version of a package, just run the below command:

pip install package_name

To install a particular version of the Package, the equal to the operator can be used

pip install PackageName==1.4

To install any package version greater than a certain version, the following command can be used

pip install PackageName>1.5

By default, pip will install the latest version of the package.