

GE8151 PROBLEM SOLVING AND PYTHON

PROGRAMMING

Unit-V

FILES, MODULES, PACKAGES

PART-A

1. Define File.

File is a collection of logically related data stored in a permanent storage such as Hard Disk. It is a Persistent storage. It can be two types: Text File and Binary File.

2. What is a text file?

A text file is a sequence of characters stored on a permanent storage medium like a hard drive, flash memory, or CD-ROM. It contains only text. Each line in text file is terminated with a special character called End of Line (EOL) character. Text files are identified with the text extension.

3. List the file opening Modes.

Files can be opened in different modes

- r→Read only mode
- r+→Read and Write mode
- rb→Read only in Binary mode
- rb+→Read and write in Binary mode
- w→Write only mode
- w+→Read and write mode
- a→Append only mode
- a+→Append and read mode

4. What are the two arguments taken by the open () function?

The open function takes two arguments: First argument – name of the file and the second argument – the mode of operation.

Example: `f=open("testdat","w")`

5. List the different ways to read a file. (or) Difference between read(), readline() and readline() methods.

To read the content of a file it must be opened in read (r mode). There are three functions available to read contents from a file. They are,

- Read() – it reads 'n' bytes in string format. If not specified it reads the entire file.
- Readlines() – it reads a line of string

- Readlines() – it reads all the lines in file and return them as list of strings.

6. What is the difference between append and write mode?

If a file is opened in append mode data will be added to the end of the existing data present in the file. But if a file is opened in write mode existing data will be removed and new data will be added to the file.

7. What is a file object?

A file object allows us to use, access and manipulate all the user accessible files. It maintains the state about the file it has opened.

Example: `f=open("test.dat","w")` //f is the file object

8. What are the attributes of file objects?

- Closed Attribute – It returns True if file state is closed, else False.
- Mode Attribute – It returns the mode in which the file is opened
- Name Attribute – It returns the name of the file which the file object is linked to.

9. List the methods in file objects.

Some of the methods used with file objects are: `open()`, `read()`, `readline()`, `readlines()`, `writelines()`, `write()`, `close()`, `seek()` and `tell()`.

10. What information is displayed if we print a file object in the given program?

If we print a file object in the given program, the name of the file, mode and the location of the object will be displayed.

Example:

```
F=open("test.txt","w")
```

```
Print(f)
```

Output:

```
<_io.TextIOWrapper name='test.txt' mode='w' encoding='cp'>
```

11. What is the use of the format operator?

The format operator % takes a format string and a tuple of expressions and yields a string that includes the expressions, formatted according to the format string.

Example:

```
Bananas = 27
```

```
Print("We have %d bananas." % Bananas)
```

Output

```
We have 27 bananas
```

12. Discover the format operator available in files.

The different format operators are,

%d,%i – Integer value

%f – floating point value

%c – character value

%s – string value.

13. What is Command Line Argument? (or) How do you use command line arguments to give input to the program?

In python, sys module provides access to any command-line arguments via sys.argv.

- Sys.argv is the list of command-line arguments.
- Len(sys.argv) is the number of command-line arguments.

Example:

```

import sys

Program_name=sys.argv[0]

Arguments = sys.argv[1:]

Count = len(arguments)
    
```

14. What is an exception?

Whenever a runtime error occurs, it creates an exception. The program stops execution and prints an error message.

Example:

```

Dividing by zero creates an exception:
Print 55/0
ZeroDivisionError:integer division or modulo
    
```

15. Differentiate Errors and Exceptions.

Errors	Exceptions
Errors are normally referred as bugs in the program. They are almost always the fault of the programmer.	An exception is an event which occurs during the execution of a program that change the normal flow of a program.
The process of finding and eliminating errors is called as debugging	Exceptions can be handled by Exception handling method using try, except.
Errors can be of: syntax errors and run time errors	Examples: ZeroDivisionError, NameError, TyoeError.

16. Explain with example the need of exceptions.

Exception handling is a mechanism by which exceptions can be handled during run time. The main purpose is to prevent abnormal termination of the program and to customize the exception message.

17. Explain built in exceptions.

The built-in exceptions can be generated by the interpreter or built-in functions. There are several built-in exceptions in python that are raised when errors occur. All instances in python must be instances of a class that derives from Base class exception.

Examples: ArithmeticError, ImportError, IndexError, NameError, OverflowError etc.

18. How to view all the built-in exception in python.

The built-in exceptions using the local() built-in functions as follows.

Syntax: locals()('__builtins__')

This will return us a dictionary of built-in exceptions, functions and attributes.

19. What are the error messages that are displayed for the following exceptions?

- a) Accessing a non-existent list item
 - b) Accessing a key that isn't the dictionary
 - c) Trying to open a non-existent file.
- a) IndexError:list index out of range
 - b) KeyError:what
 - c) IOError:[Error 2] No such file or dictionary: 'filename'.

20. Categorize the different types of errors arises during programming.

Errors are normally referred as bugs in the program. They are almost always the fault of the programmer. The process of finding and eliminating errors is called Debugging. Errors can be syntax Error, Logical Error and Runtime Error (Exceptions).

21. Give the mechanism to handle exceptions. (or) Illustrate try-except-else.

Exception can be handled by using try-except-else.

- Try block – it contains the code that cause an exception. So it is called as Exception Generation Block.
- Except block – it will handle the generated exception. So it is called as Exception Handler.
- Else block – if exception is not generated it will get executed.

Syntax:

Try:

#Statement1

Except(Exception):

#Statement2

Else:

#Statement3

22. How does try and execute work?

The try statement executes the statements in the first block. If no exception occurs, then except statement is ignored. If an exception of type IOError occurs, it executes the statements in the except branch and then continues.

Syntax:

Try:

//try block code

Except:

//except block code

Example:

Try:

Print("Hello World")

Except:

Print("This is an error message!")

23. Discover except clause with Multiple Exceptions.

Using this multiple exception can be handled using a single except block. To achieve this exception list must be added along with except statement. If the raised exception name is present in the exception list it will be handled otherwise program will be terminated.

Syntax:

Try:

#statements

Except(Exception1, Exception2, ..., Exception N):

#statements

Else:

#statements

Else:

#statements

24. What are modules? (or) write a note on modular design.

A module is simply a file that defines one or more related functions grouped together.

To rescue the functions of a given module, we need to import the module.

Syntax import <modulename>

25. What is a package?

Packages is a way to group modules, functions, classes etc. it is a Hierarchical directory structure that consists of sub packages.

Syntax: import PackageName

Example: import math

26. Interpret the following python code:

```
>>>import os
>>>cwd=os.getcwd()
>>>print cwd
/home/dinsdale
```

In the above code getcwd() is a method present in the package os that is returns current working directory of a process. In the above code the output/home/dinsdale represents the current working directory.

27. Write a python program that counts the number of words in a file.

```
Num_words=0
f=open("test.txt","r")
for line in f:
    words = line.split()
    num_words+=len(words)
print("Number of words:")
print(num_words)
```

Output:

Number of Words: 5

28. Write a python script to display the current date and time.

```
Import datetime
Print(datetime.datetime.now())
```

Output:

2019-07-16 20:11:48.193696