SHRIMATI INDIRA GANDHI COLLEGE

Affiliated to Bharathidasan University| Nationally Accredited at 'A' Grade(3rd Cycle) by NAAC

An ISO 9001:2015 Certified Institution

Thiruchirrappalli

QUESTION BANK

PROGRAMMING IN PYTHON

DEPARTMENT OF COMPUTER SCIENCE, INFORMATION TECHNOLOGY AND COMPUTER APPLICATIONS

Prepared by,

MS. C. SHYAMALADEVI, M.C.A., M.Phil., B.Ed.,

ASST. PROF. IN COMPUTER SCIENCE,

SHRIMATI INDIRA GANDHI COLLEGE,

TIRUCHIRAPPALLI – 2

PROGRAMMING IN PYTHON

UNIT I

1. Which of the following is a feature of Python?

a) Strongly typed language

b) Static typing

c) Interpreted language

d) Compiler-based language

2. How do you run a Python script from the command line?

a) python script.py

b) run script.py

c) execute script.py

d) start script.py

3. Which of the following is a valid identifier in Python?

a) 1variable

b) variable_1

c) global

d) break

4. Which of the following is a reserved keyword in Python?

a) for

b) while

c) if

d) all of the above

5. How do you define a variable in Python?

a) var x = 5

b) x = 5

c) int x = 5

d) variable x = 5

6. What is the purpose of comments in Python?

a) They are used to improve the performance of the code.

b) They are used to make the code run faster.

c) They are used to explain the code and make it more readable.

d) They are used to add extra functionality to the code.

7. Which symbol is used for indentation in Python?

a) Spaces

b) Tabs

c) Both spaces and tabs

d) None of the above

8. How do you write a multi-line statement in Python?

a) Separate each line with a comma (,)

b) Use the backslash (\) at the end of each line

c) Enclose the statement in triple quotes (""")

d) Use a semicolon (;) at the end of each line

9. What is a suite in Python?

a) A group of multiple statements within a loop or a function

b) A Python module containing multiple functions

c) A group of reserved keywords in Python

d) A data type for storing multiple values

10. Which type of quotes can be used to define a string in Python?

a) Single quotes ('')

b) Double quotes ("")

c) Both single and double quotes

d) None of the above

11. How do you take user input in Python?

a) input()

b) get_input()

c) user_input()

d) read()

12. Which function is used to print output in Python?

a) print()

b) output()

c) display()

d) show()

13. Which operator is used for exponentiation in Python?

a) ^

b) *

c) %

**d) **

14. Which data type is used to store a sequence of characters in Python?

a) Integer

b) Float

c) String

d) Boolean

15. How do you convert a string to an integer in Python?

a) str()

b) int()

c) float()

d) bool()

Answers:

1. c) Interpreted language

2. a) python script.py

3. b) variable_1

4. d) all of the above

5. b) x = 5

6. c) They are used to explain the code and make it more readable.

7. a) Spaces

8. b) Use the backslash (\) at the end of each line

9. a) A group of multiple statements within a loop or a function

10.c) Both single and double quotes

11. a) input()

12. a) print()

13. d) **

14. c) String

15. b) int()

---

1. Which of the following is an example of a Python numeric data type?

a) list

b) tuple

c) string

d) integer

2. How do you define a string in Python?

a) By enclosing it in single quotes (' ')

b) By enclosing it in double quotes (" ")

c) By enclosing it in triple quotes (''' ''')

d) All of the above

3. Which of the following is an example of a mutable data type in Python?

a) string

b) tuple

c) set

d) dictionary

4. What will be the output of the following code snippet?

```
my_list = [1, 2, 3]
print(my_list[1])
```

a) 1

b) 2

c) 3

d) Error

5. Which data type in Python represents an ordered collection of elements that can be indexed and changed?

a) tuple

b) set

c) dictionary

d) list

6. What is the result of the following code?

```
my_string = "Hello, World!"
print(len(my_string))
```

a) 6

b) 12

c) 13

d) Error

7. Which of the following data types does not allow duplicate values in Python?

a) list

b) set

c) dictionary

d) tuple

8. How do you access the value associated with a specific key in a dictionary?

a) By using the index number

b) By using the key itself

c) By using the value itself

d) By using the dictionary length

9. What will be the output of the following code snippet?

```
my_dict = {'a': 1, 'b': 2, 'c': 3}
print(my_dict.get('d'))
```

a) 1

b) 2

c) 3

d) None

10. Which function is used to convert the data type of a value in Python?

a) type()

b) convert()

c) cast()

d) str()

11. How do you add elements to a set in Python?

a) Using the append() method

b) Using the add() method

c) Using the insert() method

d) Using the extend() method

12. Which of the following is not a valid way to remove an element from a list in Python?

a) Using the remove() method

b) Using the pop() method

c) Using the delete() function

d) Using the del keyword

13. What will be the output of the following code snippet?

```
my_tuple = (1, 2, 3)
print(my_tuple[1:])
```

a) (1, 2)

b) (2, 3)

c) (1, 2, 3)

d) Error

14. Which operator is used to merge two dictionaries in Python?

a) +

b) -

c) *

d) |

15. How do you check if a key exists in a dictionary?

a) By using the in keyword

b) By using the exists() function

c) By using the contains() method

d) By using the has_key() method

Answers:

1. d) integer

2. d) All of the above

3. c) set

4. b) 2

5. d) list

6. c) 13

7. b) set

8. b) By using the key itself

9. d) None

10. c) cast()

11. b) Using the add() method

12. c) Using the delete() function

13. b) (2, 3)

14. d) |

15. a) By using the in keyword

UNIT II

Flow Control, Decision Making:

1. Which keyword is used to perform decision-making in Python?

a) for

b) while

c) if

d) switch

2. What is the output of the following code snippet?

```
x = 5
if x > 3:
    print("Greater than 3")
else:
    print("Less than or equal to 3")
```

a) Greater than 3

b) Less than or equal to 3

c) Invalid syntax

d) No output

3. What is the purpose of the "else" statement in an "if-else" block?

a) To execute a block of code if the condition is true

b) To execute a block of code if the condition is false

c) To repeat a block of code multiple times

d) To skip the current iteration of a loop

4. Which operator is used to combine multiple conditions in a Python if statement?

a) &&

b) ||

c) &

d) |

Loops, Nested Loops, Types of Loops:

5. Which loop executes a block of code a fixed number of times?

a) for loop

b) while loop

c) do-while loop

d) switch loop

6. What is the output of the following code snippet?

```
for i in range(3):
    print(i)
```

a) 0 1 2

b) 1 2 3

c) 3 2 1

d) 2 1 0

7. How do you exit a loop prematurely in Python?

a) continue

b) break

c) return

d) pass

8. What is the result of the following code?

```
for i in range(2):
    for j in range(2):
        print(i, j)
```

a) 0 0, 0 1, 1 0, 1 1

b) 0 0, 1 1

c) 0 0, 0 1

d) 1 1

9. Which loop is executed at least once, even if the condition is initially false?

a) for loop

b) while loop

c) do-while loop

d) nested loop

10. Which keyword is used to skip the current iteration of a loop?

a) skip

b) pass

c) continue

d) break

Functions: Function Definition, Calling, Arguments:

11. What is a function in Python?

a) A reserved keyword

b) A variable that stores a value

c) A block of reusable code that performs a specific task

d) A loop that repeats a set of statements

12. How do you define a function in Python?

a) def my_function():

b) function my_function():

c) define my_function():

d) func my_function():

13. What is the output of the following code snippet?

```
def greet(name):
    print("Hello, " + name)
greet("Alice")
```

a) Hello, Alice

b) Hello,

c) Error

d) No output

14. How do you call a function in Python?

a) function_name()

b) call function_name()

c) invoke function_name()

d) function_name

15. What is the purpose of function arguments in Python?

a) To specify the return value of a function

b) To define local variables within a function

c) To pass values to a function for its operation

d) To execute a block of code repeatedly

16. What is a recursive function in Python?

a) A function that calls itself

b) A function that returns a Boolean value

c) A function that performs mathematical operations

d) A function that has multiple return values

17. What is the output of the following code snippet?

```
def factorial(n):
    if n == 0:
        return 1
    else:
        return n * factorial(n-1)
print(factorial(5))
```

a) 1

b) 5

c) 10

d) 120

18. How do you define a function with multiple return values in Python?

a) Specify the return values separated by commas

b) Use the return keyword multiple times

c) Enclose the return values in square brackets

d) It is not possible to have multiple return values in Python

19. What is the purpose of the "return" statement in a function?

a) To end the execution of a function

b) To return a value from a function

c) To print output from a function

d) To define a local variable within a function

20. What is the scope of a variable defined inside a function?

a) Global scope

b) Local scope within the function

c) Both global and local scope

d) It is not possible to define variables inside a function

Recursive Functions:

21. What is the base case in a recursive function?

a) The initial condition that ends the recursion

b) The final output of the recursive function

c) The maximum number of recursive calls allowed

d) The condition for starting the recursion

22. What is the output of the following code snippet?

```
def countdown(n):
    if n <= 0:
        return
    print(n)
    countdown(n-1)
countdown(5)
```

a) 5 4 3 2 1

b) 1 2 3 4 5

c) 0

d) Error

23. What is the purpose of a recursive function?

a) To perform repetitive tasks efficiently

b) To solve complex problems by breaking them down into smaller subproblems

c) To create infinite loops

d) To execute a block of code repeatedly

24. Which of the following is an example of a recursive algorithm?

a) Linear search

b) Binary search

c) Bubble sort

d) Selection sort

Function with More than One Return Value:

25. Can a Python function return more than one value simultaneously?

a) Yes

b) No

c) Only if the return values are of the same data type

d) Only if the return values are stored in a list or tuple

26. What is the output of the following code snippet?

```
def square_and_cube(x):
    return x**2, x**3
result = square_and_cube(2)
print(result)
```

a) (4, 8)

b) [4, 8]

c) (4)

d) [4]

27. How do you access individual elements from the return value of a function with multiple return values?

a) By using the dot operator

b) By using square brackets and index number

c) By converting the return value to a string

d) By using the len() function

28. What is the output of the following code snippet?

```
def calculate(x, y):
    return x+y, x-y, x*y
result = calculate(10, 5)
print(result[2])
```

a) 15

b) 5

c) 50

d) Error

29. Can a Python function have multiple return statements?

a) Yes

b) No

c) Only if the return values are of the same data type

d) Only if the return statements are nested

30. What is the output of the following code snippet?

```
def divide(x, y):
    if y == 0:
        return "Cannot divide by zero"
    else:
        return x/y
result = divide(10, 2)
print(result)
```

a) 5

b) 10/2

c) "Cannot divide by zero"

d) None

Flow Control, Decision Making:

1. c) if

2. a) Greater than 3

3. b) To execute a block of code if the condition is false

4. d) |

Loops, Nested Loops, Types of Loops:

5. a) for loop

6. a) 0 1 2

7. b) break

8. a) 0 0, 0 1, 1 0, 1 1

9. c) do-while loop

10. c) continue

Functions: Function Definition, Calling, Arguments:

11. c) A block of reusable code that performs a specific task

12. a) def my_function():

13. a) Hello, Alice

14. a) function_name()

15. c) To pass values to a function for its operation

16. a) A function that calls itself

17. d) 120

18. a) Specify the return values separated by commas

19. b) To return a value from a function

20. b) Local scope within the function

Recursive Functions:

21. a) The initial condition that ends the recursion

22. a) 5 4 3 2 1

23. b) To solve complex problems by breaking them down into smaller subproblems

24. b) Binary search

Function with More than One Return Value:

25. a) Yes

26. a) (4, 8)

27. b) By using square brackets and index number

28. c) 50

29. a) Yes

30. a) 5

UNIT III

1. Modules and Packages:

a) What is a module in Python?

1. A built-in Python function

2. A collection of related classes and functions

3. A way to import external libraries

4. A file containing Python code that can be imported and used

2. Built-in Modules:

a) Which module provides mathematical functions and constants in Python?

1. `math`

2. `random`

3. `os`

4. `sys`

3. Creating Modules:

a) How do you import a module named `my_module` in Python?

1. **`import my_module`**
2. **`import module.my_module`**
3. **`from my_module import *`**
4. **`import my_module as mm`**

4. Locating Modules:

a) What does the `sys.path` variable represent in Python?

1. **The path of the Python interpreter**
2. **The current working directory**
3. **The list of directories Python searches for modules**
4. **The path to the Python standard library**

5. Namespaces and Scope:

a) What is a namespace in Python?

1. **A region where a variable is stored**
2. **A keyword used for defining classes**
3. **A unique identifier for a variable**
4. **A container for holding variables and functions**

6. The dir() function:

a) What does the `dir()` function do in Python?

1. Lists all available modules in Python

2. Returns the names in the current local scope or a module's attributes

3. Lists all built-in functions in Python

4. Checks if a variable is defined in the local scope

7. The reload() function:

a) Which module should be imported to use the `reload()` function in Python 3.x?

1. `reload`

2. `importlib`

3. `imp`

4. `sys`

8. Packages in Python:

a) What is a package in Python?

1. A single Python file

2. A collection of related modules

3. A built-in Python module

4. A way to install external libraries

9. Date and Time Modules:

a) Which module provides date and time functionalities in Python?

1. **`time`**
2. **`datetime`**
3. **`calendar`**
4. **`date`**

10. File Handling:

a) How do you open a file named "example.txt" in read mode in Python?

1. **`file = open("example.txt", "r")`**
2. **`file = open("example.txt", mode="r")`**
3. **`file = open("example.txt", read=True)`**
4. **`file = open("example.txt", mode="read")`**

11. Directories in Python:

a) How do you check if a directory exists in Python?

1. **Use the `exists()` function from the `os` module**
2. **Use the `isdir()` function from the `os.path` module**
3. **Use the `dir_exists()` function from the `os` module**
4. **Use the `exists()` function from the `os.path` module**

12. Modules and Packages:

a) What is the purpose of a package in Python?

1. To organize modules and provide a hierarchical structure

2. To import external libraries

3. To create a single Python file

4. To define built-in functions

13. Built-in Modules:

a) Which module is used for interacting with the operating system in Python?

1. `os`

2. `sys`

3. `math`

4. `datetime`

14. Namespaces and Scope:

a) What is the difference between global and local scope in Python?

1. Global scope refers to variables defined within a function, while local scope refers to variables defined outside any function.

2. Global scope refers to variables defined outside any function, while local scope refers to variables defined within a function.

3. Global scope refers to variables defined within a module, while local scope refers to variables defined within a class.

4. Global scope refers to variables defined within a class, while local scope refers to variables defined within a module.

15. File Handling:

a) How do you close a file in Python after reading or writing?

1. `file.close()`

2. `close(file)`

3. `file.exit()`

4. `exit(file)`

1. Modules and Packages:

a) Answer: 4. A file containing Python code that can be imported and used

2. Built-in Modules:

a) Answer: 1. `math`

3. Creating Modules:

a) Answer: 4. `import my_module as mm`

4. Locating Modules:

a) Answer: 3. The list of directories Python searches for modules

5. Namespaces and Scope:

a) Answer: 4. A container for holding variables and functions

6. The dir() function:

a) Answer: 2. Returns the names in the current local scope or a module's attributes

7. The reload() function:

a) Answer: 2. `importlib`

8. Packages in Python:

a) Answer: 2. A collection of related modules

9. Date and Time Modules:

a) Answer: 2. `datetime`

10. File Handling:

a) Answer: 1. `file = open("example.txt", "r")`

11. Directories in Python:

a) Answer: 2. Use the `isdir()` function from the `os.path` module

12. Modules and Packages:

a) Answer: 1. To organize modules and provide a hierarchical structure

13. Built-in Modules:

a) Answer: 1. `os`

14. Namespaces and Scope:

a) Answer: 2. Global scope refers to variables defined outside any function, while local scope refers to variables defined within a function.

15. File Handling:

a) Answer: 1. `file.close()`

1. Modules and Packages:

a) What is the difference between a module and a package in Python?

1. A module is a single Python file, while a package is a collection of modules.

2. A module contains functions, while a package contains classes.

3. A module is used for file handling, while a package is used for mathematical operations.

4. A module is created by the user, while a package is provided by Python's standard library.

2. Built-in Modules:

a) Which module provides functionalities for working with regular expressions in Python?

1. `re`

2. `os`

3. `sys`

4. `math`

3. Creating Modules:

a) How do you create a module in Python?

1. By saving a Python script with a `.mod` extension

2. By creating a directory with the module name and placing Python scripts inside it

3. By defining a class and using it as a module

4. By importing a built-in module and modifying it

4. Locating Modules:

a) What is the purpose of the `PYTHONPATH` environment variable in Python?

1. It defines the location of the Python interpreter on the system.

2. It specifies the default directory for storing Python scripts.

3. It sets the search path for modules when importing.

4. It determines the file extension for Python modules.

5. Namespaces and Scope:

a) What is the scope of a variable defined inside a function in Python?

1. **Local scope**
2. **Global scope**
3. **Class scope**
4. **Built-in scope**

6. The dir() function:

a) Which of the following statements is true about the `dir()` function in Python?

1. **It lists all available modules in Python.**
2. **It lists all built-in functions and variables in Python.**
3. **It returns a sorted list of attributes of an object or a module.**
4. **It displays the current working directory.**

7. The reload() function:

a) Which module should be imported to use the `reload()` function in Python 2.x?

1. **`reload`**
2. **`importlib`**
3. **`imp`**
4. **`sys`**

8. Packages in Python:

a) How are modules organized within a package in Python?

1. By placing them in a directory with the package name and importing them.

2. By defining them as classes and using them as packages.

3. By including them in a separate file with the package name.

4. By saving them as individual scripts in the package directory.

9. Date and Time Modules:

a) Which module provides functionalities for formatting dates and times in Python?

1. `datetime`

2. `time`

3. `calendar`

4. `date`

10. File Handling:

a) What is the purpose of the `with` statement in file handling in Python?

1. It opens a file in write mode.

2. It automatically closes the file after reading or writing.

3. It reads the entire file content at once.

4. It provides an interface for interacting with the operating system.

11. Directories in Python:

a) Which module is used for interacting with directories in Python?

1. **`os`**
2. **`sys`**
3. **`shutil`**
4. **`pathlib`**

12. Modules and Packages:

a) How do you import a module named `my_module` in Python using the `from` statement?

1. **`from my_module import module`**
2. **`from my_module import my_module`**
3. **`from my_module import *`**
4. **`from my_module import function`**

13. Built-in Modules:

a) Which module is used for working with command-line arguments in Python?

1. **`argparse`**
2. **`sys`**
3. **`os`**
4. **`math`**

14. Namespaces and Scope:

a) What happens if you try to access a variable defined inside a function outside its scope?

1. It will raise an error.

2. It will be accessible and have the same value as inside the function.

3. It will be accessible but have a different value than inside the function.

4. It will only be accessible in the global scope.

15. File Handling - Directories in Python:

a) Which method is used to check if a directory exists in Python?

1. `os.path.exists()`

2. `os.path.isfile()`

3. `os.path.isdir()`

4. `os.path.getsize()`

1. Modules and Packages:

a) Answer: 1. A module is a single Python file, while a package is a collection of modules.

2. Built-in Modules:

a) Answer: 1. `re`

3. Creating Modules:

a) Answer: 2. By creating a directory with the module name and placing Python scripts inside it

4. Locating Modules:

a) Answer: 3. It sets the search path for modules when importing.

5. Namespaces and Scope:

a) Answer: 1. Local scope

6. The dir() function:

a) Answer: 3. It returns a sorted list of attributes of an object or a module.

7. The reload() function:

a) Answer: 3. `imp`

8. Packages in Python:

a) Answer: 1. By placing them in a directory with the package name and importing them.

9. Date and Time Modules:

a) Answer: 1. `datetime`

10. File Handling:

a) Answer: 2. It automatically closes the file after reading or writing.

11. Directories in Python:

a) Answer: 1. `os`

12. Modules and Packages:

a) Answer: 3. `from my_module import *`

13. Built-in Modules:

a) Answer: 1. `argparse`

14. Namespaces and Scope:

a) Answer: 1. It will raise an error.

15. File Handling:

a) Answer: 1. `os.path.isdir()` is used to check if a directory exists in Python.

UNIT IV

1. Object-Oriented Programming:

a) What is Object-Oriented Programming (OOP)?

1. A programming paradigm that focuses on data structures

2. A programming style that emphasizes functions

3. A programming approach based on objects and classes

4. A programming technique used in low-level programming languages

2. Class Definition:

a) What is a class in Python?

1. A function that performs a specific task

2. A built-in data structure

3. A template for creating objects

4. A module that contains related functions

3. Creating Objects:

a) How do you create an object of a class in Python?

1. Using the `new` keyword

2. By calling the class as a function

3. Using the `create` method

4. By importing the class from a module

4. Built-in Attribute Methods:

a) Which built-in attribute method returns the string representation of an object?

1. `__str__()`

2. `__init__()`

3. `__repr__()`

4. `__len__()`

5. Built-in Class Attributes:

a) Which built-in class attribute returns the name of the class as a string?

1. `__name__`

2. `__class__`

3. `__doc__`

4. `__module__`

6. Destructors in Python:

a) What is a destructor in Python?

1. A method that initializes an object

2. A special method that is automatically called when an object is destroyed

3. A function that destroys an object manually

4. A built-in Python function for memory management

7. Encapsulation:

a) What is encapsulation in OOP?

1. The process of hiding sensitive data within an object

2. The process of exposing all data publicly

3. The process of defining multiple classes in a single file

4. The process of creating objects from a class

8. Data Hiding:

a) How is data hidden in Python?

1. By using private variables and methods with names starting with double underscores

2. By using global variables

3. By using public variables and methods with names starting with a single underscore

4. By defining classes inside functions

9. Inheritance:

a) What is inheritance in OOP?

1. The process of creating multiple objects from a single class

2. The process of reusing code and creating a new class from an existing class

3. The process of defining attributes and methods in a class

4. The process of hiding data within an object

10. Method Overriding:

a) What is method overriding in Python?

1. The process of creating multiple methods with the same name in a class

2. The process of modifying the behavior of an inherited method in a subclass

3. The process of calling a method from another method in the same class

4. The process of defining methods inside another method

11. Polymorphism:

a) What is polymorphism in OOP?

1. The process of defining multiple classes in a single file

2. The process of creating objects from a class

3. The process of hiding data within an object

4. The ability of an object to take on many forms and respond differently based on the context

12. Object-Oriented Programming:

a) What is the main advantage of Object-Oriented Programming?

1. Code reusability

2. Simplicity of syntax

3. Better memory management

4. Faster execution speed

13. Class Definition:

a) What is the `self` parameter in Python class methods?

1. It refers to the class itself

2. It refers to the object being created

3. It is a keyword that is used to define classes

4. It is a reserved word and has no specific meaning

14. Creating Objects:

a) Which keyword is used to create an object in Python?

1. `new`

2. `create`

3. `instantiate`

4. No keyword is required

15. Built-in Attribute Methods:

a) Which built-in attribute method is called when an object is created from a class?

1. `__init__()`

2. `__str__()`

3. `__repr__()`

4. `__len__()`

16. Built-in Class Attributes:

a) What is the purpose of the `__name__` attribute in Python classes?

1. It stores the name of the class

2. It stores the documentation string of the class

3. It stores the module name of the class

4. It stores the superclass of the class

17. Destructors in Python:

a) Which special method is used as a destructor in Python?

1. `__init__()`

2. `__del__()`

3. `__str__()`

4. `__repr__()`

18. Encapsulation:

a) What is encapsulation in OOP?

1. The process of hiding sensitive data within an object

2. The process of exposing all data publicly

3. The process of defining multiple classes in a single file

4. The process of creating objects from a class

19. Data Hiding:

a) How is data hidden in Python?

1. By using private variables and methods with names starting with double underscores

2. By using global variables

3. By using public variables and methods with names starting with a single underscore

4. By defining classes inside functions

20. Inheritance:

a) What is inheritance in OOP?

1. The process of creating multiple objects from a single class

2. The process of reusing code and creating a new class from an existing class

3. The process of defining attributes and methods in a class

4. The process of hiding data within an object

21. Method Overriding:

a) What is method overriding in Python?

1. The process of creating multiple methods with the same name in a class

2. The process of modifying the behavior of an inherited method in a subclass

3. The process of calling a method from another method in the same class

4. The process of defining methods inside another method

22. Polymorphism:

a) What is polymorphism in OOP?

1. The process of defining multiple classes in a single file

2. The process of creating objects from a class

3. The process of hiding data within an object

4. The ability of an object to take on many forms and respond differently based on the context

23. Object-Oriented Programming:

a) What is the main advantage of Object-Oriented Programming?

1. Code reusability

2. Simplicity of syntax

3. Better memory management

4. Faster execution speed

24. Class Definition:

a) What is the `self` parameter in Python class methods?

1. It refers to the class itself

2. It refers to the object being created

3. It is a keyword that is used to define classes

4. It is a reserved word and has no specific meaning

25. Creating Objects:

a) Which keyword is used to create an object in Python?

1. `new`

2. `create`

3. `instantiate`

4. No keyword is required

26. Built-in Attribute Methods:

a) Which built-in attribute method is called when an object is created from a class?

1. `__init__()`

2. `__str__()`

3. `__repr__()`

4. `__len__()`

27. Built-in Class Attributes:

a) What is the purpose of the `__name__` attribute in Python classes?

1. It stores the name of the class

2. It stores the documentation string of the class

3. It stores the module name of the class

4. It stores the superclass of the class

28. Destructors in Python:

a) What is the purpose of a destructor in Python?

1. To initialize an object

2. To perform clean-up actions before an object is destroyed

3. To create multiple objects from a single class

4. To define methods and attributes within a class

29. Encapsulation:

a) Which concept in OOP allows bundling of data and methods into a single unit?

1. Inheritance

2. Encapsulation

3. Polymorphism

4. Abstraction

30. Polymorphism:

a) What is polymorphism in Python?

1. The process of creating multiple objects from a single class

2. The ability of an object to take on many forms and respond differently based on the context

3. The process of defining multiple classes in a single file

4. The process of modifying the behavior of an inherited method in a subclass

1. Object-Oriented Programming:

a) Answer: 3. A programming approach based on objects and classes

2. Class Definition:

a) Answer: 3. A template for creating objects

3. Creating Objects:

a) Answer: 2. By calling the class as a function

4. Built-in Attribute Methods:

a) Answer: 1. `__str__()`

5. Built-in Class Attributes:

a) Answer: 1. `__name__`

6. Destructors in Python:

a) Answer: 2. `__del__()`

7. Encapsulation:

a) Answer: 1. The process of hiding sensitive data within an object

8. Data Hiding:

a) Answer: 1. By using private variables and methods with names starting with double underscores

9. Inheritance:

a) Answer: 2. The process of reusing code and creating a new class from an existing class

10. Method Overriding:

a) Answer: 2. The process of modifying the behavior of an inherited method in a subclass

11. Polymorphism:

a) Answer: 4. The ability of an object to take on many forms and respond differently based on the context

12. Object-Oriented Programming:

a) Answer: 1. Code reusability

13. Class Definition:

a) Answer: 2. It refers to the object being created

14. Creating Objects:

a) Answer: 4. No keyword is required

15. Built-in Attribute Methods:

a) Answer: 1. `__init__()`

16. Built-in Class Attributes:

a) Answer: 3. It stores the module name of the class

17. Destructors in Python:

a) Answer: 2. `__del__()`

18. Encapsulation:

a) Answer: 1. The process of hiding sensitive data within an object

19. Data Hiding:

a) Answer: 1. By using private variables and methods with names starting with double underscores

20. Inheritance:

a) Answer: 2. The process of reusing code and creating a new class from an existing class

21. Method Overriding:

a) Answer: 2. The process of modifying the behavior of an inherited method in a subclass

22. Polymorphism:

a) Answer: 4. The ability of an object to take on many forms and respond differently based on the context

23. Object-Oriented Programming:

a) Answer: 1. Code reusability

24. Class Definition:

a) Answer: 2. It refers to the object being created

25. Creating Objects:

a) Answer: 4. No keyword is required

26. Built-in Attribute Methods:

a) Answer: 1. `__init__()`

27. Built-in Class Attributes:

a) Answer: 1. It stores the name of the class

28. Destructors in Python:

a) Answer: 2. To perform clean-up actions before an object is destroyed

29. Encapsulation:

a) Answer: 2. Encapsulation

30. Polymorphism:

a) Answer: 2. The ability of an object to take on many forms and respond differently based on the context

UNIT V

1. Which keyword is used to handle exceptions in Python?

a) try

b) catch

c) except

d) handle

2. Which of the following is NOT a built-in exception in Python?

a) ValueError

b) TypeError

c) FileNotFoundException

d) IndexError

3. What is the purpose of exception handling in Python?

a) To ignore errors and continue execution

b) To handle unexpected situations gracefully

c) To terminate the program immediately

d) To increase the speed of program execution

4. Which statement is used to raise an exception manually in Python?

a) throw

b) catch

c) assert

d) raise

5. What is a user-defined exception in Python?

a) An exception raised by the user

b) An exception defined by the Python programming language

c) An exception that is not predefined and can be created by the user

d) An exception raised when the program encounters an error

6. Which keyword is used to define assertions in Python?

a) check

b) assert

c) verify

d) validate

7. What is the purpose of assertions in Python?

a) To handle runtime errors

b) To stop the program execution

c) To provide debugging information

d) To check if a given condition is true

8. Which function is used to match a pattern at the beginning of a string using regular expressions?

a) findall()

b) match()

c) search()

d) replace()

9. Which function is used to search for a pattern anywhere in a string using regular expressions?

a) findall()

b) match()

c) search()

d) replace()

10. Which function is used to replace occurrences of a pattern in a string using regular expressions?

a) findall()

b) match()

c) search()

d) replace()

11. Which modifier is used to perform case-insensitive matching in regular expressions?

a) i

b) g

c) m

d) s

12. What are regular expression patterns?

a) Strings used to represent regular expressions

b) Functions used to apply regular expressions

c) Libraries used to handle regular expressions

d) Variables used to store regular expressions

13. Which character class matches any digit (0-9) in regular expressions?

a) \w

b) \s

c) \d

d) \b

14. Which character class matches any whitespace character in regular expressions?

a) \w

b) \s

c) \d

d) \b

15. Which character class matches any word character (alphanumeric and underscore) in regular expressions?

a) \w

b) \s

c) \d

d) \b

16. What does the repetition modifier '+' do in regular expressions?

a) Matches zero or one occurrence of the preceding pattern

b) Matches one or more occurrences of the preceding pattern

c) Matches zero or more occurrences of the preceding pattern

d) Matches a specific number of occurrences of the preceding pattern

17. Which method is used to find all occurrences of a pattern in a string using regular expressions?

a) findall()

b) match()

c) search()

d) replace()

18. Which method is used to compile a regular expression pattern into a pattern object?

a) findall()

b) match()

c) search()

d) compile()

19. True or False: Regular expressions can only be used to search for patterns in text.

a) True

b) False

20. True or False: Regular expressions are case-insensitive by default.

a) True

b) False

21. True or False: Regular expressions can be used to validate email addresses.

a) True

b) False

22. True or False: Regular expressions can be used to extract information from HTML tags.

a) True

b) False

23. True or False: Regular expressions can be used to replace specific parts of a string with another value.

a) True

b) False

24. True or False: Regular expressions can be used to split a string into a list based on a pattern.

a) True

b) False

25. True or False: Regular expressions can match patterns that occur at the beginning of a line.

a) True

b) False

26. True or False: Regular expressions can match patterns that occur at the end of a line.

a) True

b) False

27. True or False: Regular expressions can match patterns that span multiple lines.

a) True

b) False

28. True or False: Regular expressions can match patterns that contain multiple characters.

a) True

b) False

29. True or False: Regular expressions can match patterns that include special characters.

a) True

b) False

30. True or False: Regular expressions can be used to validate phone numbers.

a) True

b) False

Here are the answers to the MCQs:

1. c) except

2. c) FileNotFoundException

3. b) To handle unexpected situations gracefully

4. d) raise

5. c) An exception that is not predefined and can be created by the user

6. b) assert

7. d) To check if a given condition is true

8. b) match()

9. c) search()

10. d) replace()

11. a) i

12. a) Strings used to represent regular expressions

13. c) \d

14. b) \s

15. a) \w

16. b) Matches one or more occurrences of the preceding pattern

17. a) findall()

18. d) compile()

19. b) False

20. b) False

21. a) True

22. a) True

23. a) True

24. a) True

25. a) True

26. b) False

27. a) True

28. a) True

29. a) True

30. a) True