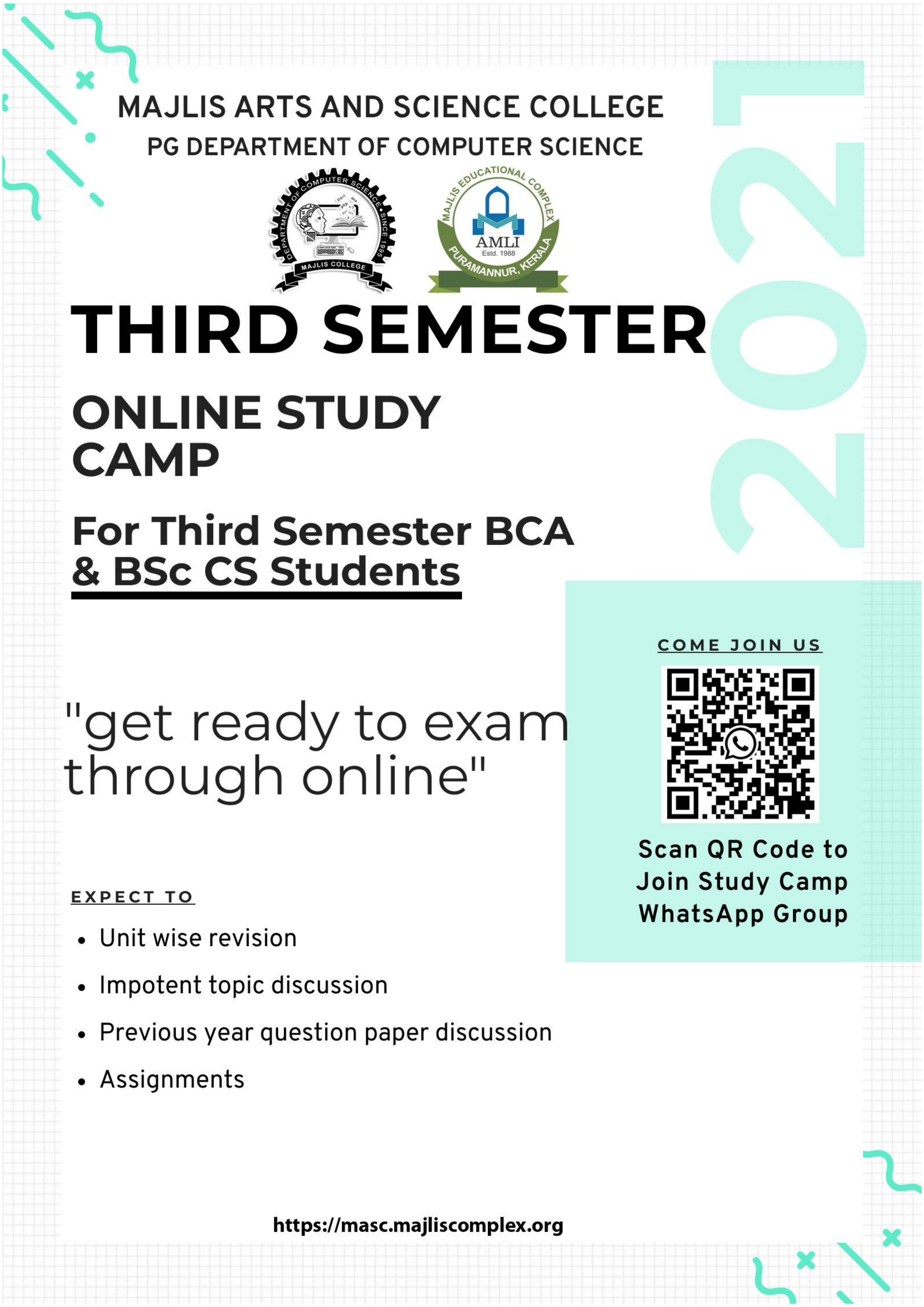


MAJLIS ARTS AND SCIENCE COLLEGE

PG DEPARTMENT OF COMPUTER SCIENCE

THIRD SEMESTER

ONLINE STUDY CAMP

For Third Semester BCA & BSc CS Students

"get ready to exam through online"

COME JOIN US

Scan QR Code to Join Study Camp WhatsApp Group

EXPECT TO

- Unit wise revision
- Impotent topic discussion
- Previous year question paper discussion
- Assignments

https://masc.majliscomplex.org

Python Programming

Module-1

1. **Python is a Free and Open Source language. What do you understand by this feature?**

   It means Python is freely available without any cost. Its source code is also available. One can modify, improve/extend open-source software.

2. **What is Python?**

   Python is a high-level, interpreted, interactive and object-oriented scripting language. Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

3. **What is the difference between Interactive mode and Script Mode in Python?**

   In interactive mode, one command can run at a time and commands are not saved. Whereas in Script mode, we can save all the commands in the form of a program file and can see output of all lines together.

4. **What is the difference between a keyword and an identifier?**

   Keywords are predefined reserved words, which possess special meaning. An identifier is a unique name given to a particular variable, function or class in the program.

5. **What is the difference between an expression and a statement in Python?**

   A statement is an instruction that the Python interpreter can execute. An expression is a combination of values, variables, operators, and calls to functions. Expressions need to be evaluated.

6. **What are tokens in Python? How many types of tokens allowed in Python?**

   Tokens are the smallest unit of the program. There are following tokens in Python:

   - Reserved words or Keywords
   - Identifiers
   - Literals
   - Operators
   - Punctuators

7. **What is IDLE?**

   IDLE is Python's Integrated Development and Learning Environment. It allows programmers to easily write Python code. IDLE can be used to execute a single statement and create, modify, and execute Python scripts. IDLE provides a fully-featured text editor to create Python scripts that include features like syntax highlighting, auto completion, and smart indent.

8. How to delete a variable in python?

Python provides a feature to delete a variable when it is not in use so as to free up space. Using the command del ‘variable name’, we can delete any specific variable.
Example
del a

9. Explain Python interpreter

Python is an interpreter language. It means it executes the code line by line. Python provides a Python Shell, which is used to execute a single Python command and display the result.

10. What is the use of type() function

In Python data types are classes and variables are the object of these classes. To determine what type of object declared in the program there is a built-in function in Python known as type(). The type() **function** in Python returns the data type of the object passed to it as an argument. This function is very useful in the debugging process.

11. Differentiate between Type Conversion and Type Casting

Type Conversion is the conversion of object from one data type to another data type. Implicit Type Conversion is automatically performed by the Python interpreter. Explicit Type Conversion is also called Type Casting, the data types of objects are converted using predefined functions by the user.

12. What is the Role of Indentation in Python?

The identification of a block of code in Python is done using Indentation. Most of the programming languages like C, C++, and Java use braces { } to define a block of code. Python, however, uses indentation. A code block (body of a function, loop, etc.) starts with indentation and ends with the first unindented line

13. What is a comment? How to write comments in python?

Using comments in programs makes our code more understandable. It makes the program more readable which helps us remember why certain blocks of code were written. Comments can also be used to ignore some code while testing other blocks of code. In Python, we use the ***hash symbol #*** to write a single-line comment. Everything that comes after # is ignored. We can use multiline strings (triple quotes) to write multiline comments.

14. What is variable in python? How to assign values to a variable?

It is a reserved memory location that stores and manipulates data. You can use the assignment operator = to assign a value to a variable. Python is a type-inferred language, so you don't have to explicitly define the variable type.
You can assign multiple values to multiple variables by separating variables and values with commas
e.g
a, b = 100, 200
You can assign the same value to multiple variables by using = consecutively.
e.g
a = b = 100

15. What are the rules that we need to keep in mind while creating Python variables?

There are some rules that we need to keep in mind while creating Python variables.

- Name of a variable cannot start with a number. It should start with either an alphabet or the underscore character.
- Variable names are always case sensitive and can contain alphanumeric characters and the underscore character.
- Reserved words cannot be used as variable names.

16. What are the two categories of statements in python?

There are two categories of statements in Python:

- **Expression Statements**
- **Assignment Statements**

With the help of expressions, we perform operations like addition, subtraction, concatenation etc. In other words, it is a statement that returns a value.
With the help of assignment statements, we create new variables, assign values and also change values.

17. Differentiate between Mutable and Immutable Types

Data are stored in a computer's memory for processing. Some of these values can be modified during processing, but contents of others can't be altered once they are created in the memory.
Numbers, strings, and Tuples are immutable, which means their contents can't be altered after creation. On the other hand, items in a List or Dictionary object can be modified. It is possible to add, delete, insert, and rearrange items in a list or dictionary. Hence, they are mutable objects.

18. What is the use of eval() function in python?

The eval() method parses the expression passed to it and runs python expression(code) within the program.

Example

```
x = 1
print(eval('x + 1'))
```

Output

```
2
```

19. Explain the features of python

- **Easy to Code**

Python is a very developer-friendly language which means that anyone and everyone can learn to code it in a couple of hours or days. The syntax rules of Python allow you to express concepts without writing additional code. Python, unlike other programming languages, emphasizes on code readability. The readable and clean code base will help you to maintain and update the software without putting extra time and effort.

- **Open Source and Free**

Python is an open-source programming language which means that anyone can create and contribute to its development. Python has an online forum where thousands of coders gather daily to improve this language further. Along with this Python is free to download and use in any operating system. As an open source programming language, Python helps you to curtail software development cost significantly. You can even use several open source Python frameworks, libraries and development tools to curtail development time without increasing development cost. you can simplify and speedup web application development by using robust Python web frameworks like Django, Flask, Pyramid, Bottle and Cherrypy.

- **Interpreted Language**

Python is an **interpreted language** i.e. interpreter executes the code line by line at a time. When you use an interpreted language like Python, there is no separate compilation and execution steps. This makes **debugging** easy and thus suitable for beginners. Internally, **Python** converts the source code into an intermediate form called **bytecodes** and then translates this into the native language of your specific computer and then runs it.

- **Cross-platform language**

Python can run equally on **different platforms** such as Windows, Linux, Unix , Macintosh etc. A Python program written on a Macintosh computer will run on a Linux system and vice versa. Thus, Python is a **portable language**. Python is an interpreted programming language. It allows you to you to run the same code on multiple platforms without recompilation.

- **Support for GUI**

GUI or Graphical User Interface is one of the key aspects of any programming language. In python you can accelerate desktop GUI application development using **Python GUI frameworks** and toolkits like PyQT, PyJs, PyGUI, Kivy, PyGTK and WxPython.

- **Object-Oriented language**

Python supports **object oriented** features. Python has all features of an object-oriented language such as inheritance, method overriding, objects, etc. Thus it supports all the paradigms and has corresponding functions in their libraries. It also supports the implementation of multiple inheritances,

- **Simplify Complex Software Development**

Python is a general purpose programming language. Hence, you can use the programming language for developing both desktop and web applications. Also, you can use Python for developing complex scientific and numeric applications. Python is designed with features to facilitate data analysis and visualization. Many **Python developers** even use Python to accomplish artificial intelligence (AI) and natural language processing tasks.

20. What are the different operators used in Python?

Python language supports the following types of operators.

1. Arithmetic Operators
2. Bitwise Operators
3. Membership Operators
4. Identity Operators
5. Comparison Operators
6. Assignment Operators
7. Logical Operators

1. Arithmetic Operator

Arithmetic operators used to perform mathematical operations

Operator	Description	Syntax	Output
+	Addition	a+b	Returns sum of the operands
–	Subtraction	a-b	Returns Difference of the operands
/	Division	a/b	Returns Quotient of the operands
*	Multiplication	a*b	Returns product of the operands

**	Exponentiation	a**b	returns exponent of a raised to the power b
%	Modulus	a%b	returns remainder of the division
//	Floor division	a//b	returns real value and ignores decimal part

2. Bitwise Operators

Refers to the operators working on bit the box below provides the bitwise operators in python

Operator	Description	Syntax	Output
&	Binary AND	a&b	copies a bit to the result if it exists in both operands
|	Binary OR	a|b	Copies a bit if it exists in either operand.
^	Binary XOR	a^b	Copies the bit if it is set in one operand but not both.
~	Binary One's Complement	a~b	Unary operation of flipping bits
<<	Binary Left Shift	a<<b	Left operands value is moved left by the number of bits specified by the right operand.
>>	Binary Right Shift	a>>b	Left operands value is moved right by the number of bits specified by the right operand.

3. Membership Operators

Refers to the operators used in validation of membership of operand test in a sequence, such as strings, lists, or tuples. There are two types of membership operators in python

Operator	Syntax	Output
in	if (a in x):	Evaluates to true if it finds a variable in the specified sequence and false otherwise.
not in	If (b not in x):	Evaluates to true if it does not finds a variable in the specified sequence and false otherwise.

4. Identity Operators

Used to compare the memory locations of the operands, they are quite often used to determine if the operand is of a particular type, there are two types of identity operators in python.

Operator	Syntax	Output
is	x is y	returns True if the type of the value in y points to the same type in the x
is not	x is not y	returns True if the type of the value in y points to a different type than the value in the x

5. Comparison Operators

Also known as Relational operators, these operators are used in determining the relation between the operand on either side of the operator.

Operator	Syntax	Output
==	(a == b)	If the values of a and b are equal, then the condition becomes true.
!=	(a != b)	If values of a and b are not equal, then condition becomes true.
<>	(a <> b)	If values of a and b are not equal, then condition becomes true.

>	(a > b)	If the value of a is greater than the value of b, then condition becomes true.
<	(a < b)	If the value of a is less than the value of b, then condition becomes true.
>=	(a >= b)	If the value of a is greater than or equal to the value of b, then condition becomes true.
<=	(a <= b)	If the value of b is less than or equal to the value of b, then condition becomes true.

6. Assignment Operators

Refer as the name suggests is used to declare assignments to the operands, following are the types of assignment operators in python.

Operator	Description	Syntax	Output
=	Equal to	c = a + b	assigns a value of a + b into c
+=	Add AND	c += a	is equivalent to c = c + a
-=	Subtract AND	c -= a	is equivalent to c = c – a
*=	Multiply AND	c *= a	is equivalent to c = c * a
/=	Divide AND	c /= a	is equivalent to c = c / ac /= a is equivalent to c = c / a

%=	Modulus AND	c %= a	is equivalent to c = c % a
**=	Exponent AND	c **= a	is equivalent to c = c ** a
//=	Floor Division	c //= a	is equivalent to c = c // a

7. Logical Operators

These operators are used to perform similar operations as that of logical gates, there are 3 types of logical operators in python.

Operator	Description	Syntax	Output
and	Logical AND	a and b	a condition is true if both a and b are true
or	Logical OR	a or b	a condition is true if either a and b are true
not	Logical NOT	not a	Complement the operand

21. Explain different data type used in python

Python provides various standard data types that define the storage method on each of them. The data types defined in Python are given below.

- Numbers
- Sequence Type
- Boolean
- Set
- Dictionary

1. Numbers

Number stores numeric values. The integer, float, and complex values belong to a Python Numbers data-type. Python supports three types of numeric data.

Int – It contains positive or negative whole numbers (without fraction or decimal). In Python there is no limit to how long an integer value can be.

Float - Float is used to store floating-point numbers like 1.9, 9.902, 15.2, etc. It is accurate upto 15 decimal points.

complex - A complex number contains an ordered pair, i.e., x + iy where x and y denote the real and imaginary parts, respectively.

2. Sequence Type

String: - The string can be defined as the sequence of characters represented in the quotation marks. In Python, we can use single, double, or triple quotes to define a string.

Example

```
A=”Python programming”
```

List: - A list object is an ordered collection of one or more data items, not necessarily of the same type. The items stored in the list are separated with a comma (,) and enclosed within square brackets [].

Example

```
L= [34, 2.5, True, ”Apple”]
```

Tuple: - A Tuple object is an ordered collection of one or more data items, not necessarily of the same type. The items of the tuple are separated with a comma (,) and enclosed in parentheses ().

Example

```
T= (25, True, 3.5,”Orange”)
```

3. Dictionary

Dictionary is an unordered set of a key-value pair of items. Key can hold any primitive data type, whereas value is an arbitrary Python object. The items in the dictionary are separated with the comma (,) and enclosed in the curly braces {}.

Example

```
D={1:”Jan”, 2:”Feb” , 3:”Mar”}
```

4. Boolean

Boolean type provides two built-in values, True and False. These values are used to determine the given statement true or false. It denotes by the class bool. True can be represented by any non-zero value or 'T' whereas false can be represented by the 0 or 'F'.

Example

```
B=True
```

5. Set

Python Set is the unordered collection of the data type. It is mutable and has unique elements. In set, the order of the elements is undefined. The set is created by using a built-in function set(), or a sequence of elements is passed in the curly braces and separated by the comma. It can contain various types of values.

Example

S=set(23,45,67)

22. Specify the area where Python can be applied.

Python is known for its general-purpose nature that makes it applicable in almost every domain of software development. Here, we are specifying application areas where Python can be applied.

Web Applications

We can use Python to develop web applications. It provides libraries to handle internet protocols such as HTML and XML etc . Python provides many useful frameworks such as Django , Flask and Bottle.

Desktop GUI Applications

The GUI stands for the Graphical User Interface, which provides a smooth interaction to any application. Python provides a Tk GUI library to develop a user interface

Console-based Application

Console-based applications run from the command-line or shell. These applications are computer program which are used commands to execute. Python can develop this kind of application very effectively. It is famous for having REPL, which means the Read-Eval-Print Loop that makes it the most suitable language for the command-line applications.

Scientific and Numeric

Python language is the most suitable language for Artificial intelligence or machine learning. It consists of many scientific and mathematical libraries, which makes easy to solve complex calculations.

Business Applications

E-commerce and ERP are an example of a business application. This kind of application requires extensively, scalability and readability, and Python provides all these features. Python provides a Tryton platform which is used to develop the business application.

Audio or Video-based Applications

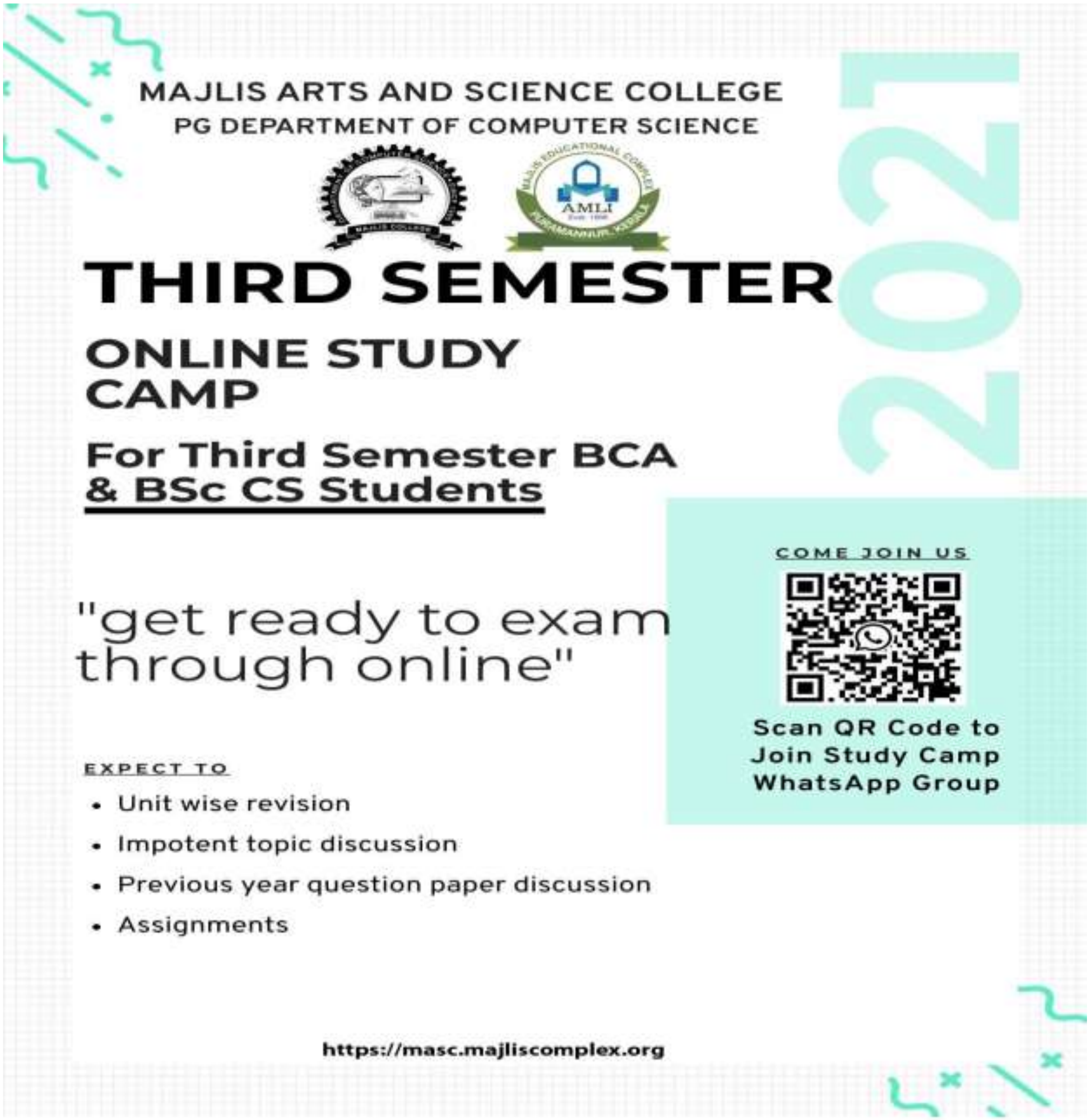
Python is flexible to perform multiple tasks and can be used to create multimedia applications. The few multimedia libraries are Pyglet ,QT Phonon etc.

3D CAD Applications

The CAD (Computer-aided design) is used to design engineering related architecture. It is used to develop the 3D representation of a part of a system. Python can create a 3D CAD application

Enterprise Applications

Python can be used to create applications that can be used within an Enterprise or an organization. Some real-time applications are OpenERP, Tryton, Picalo, etc.



MAJLIS ARTS AND SCIENCE COLLEGE

PG DEPARTMENT OF COMPUTER SCIENCE

2021

THIRD SEMESTER

ONLINE STUDY CAMP

For Third Semester BCA & BSc CS Students

"get ready to exam through online"

COME JOIN US

Scan QR Code to Join Study Camp WhatsApp Group

EXPECT TO

- Unit wise revision
- Impotent topic discussion
- Previous year question paper discussion
- Assignments

https://masc.majliscomplex.org

BCA AND BSC COMPUTER SCIENCE

PYTHON PROGRAMMING

MODULE 2

1.Explain IF statement in Python?

IF statement will execute block of statements only if the condition is true.

Syntax:

```
if test expression:
   statement(s)
```

Here, the program evaluates the test expression and will execute statement(s) only if the test expression is True. If the test expression is False, the statement(s) is not executed.

In Python, the body of the if statement is indicated by the indentation. The body starts with an indentation and the first unindented line marks the end.

Python interprets non-zero values as True. None and 0 are interpreted as False.

Python if Statement Flowchart

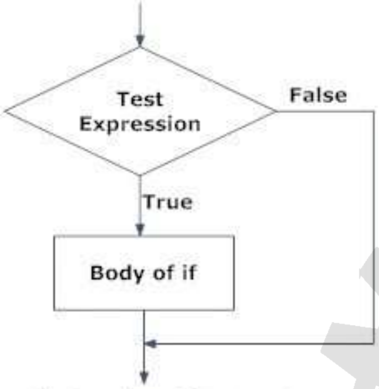


Fig: Operation of if statement

Example

```
num = 3
if num > 0:
   print(num, "is a positive number.")
```

2.Explain IF … ELSE statement?

The **if-else statement** checks the expression and **executes** the **if block** when the expression is **True** otherwise it will **execute** the **else block of code**.

Syntax:

```
if test expression:
   Body of if
else:
   Body of else
```

The if..else statement evaluates test expression and will execute the body of if only when the test condition is True. If the condition is False, the body of else is executed. Indentation is used to separate the blocks.

Flowchart

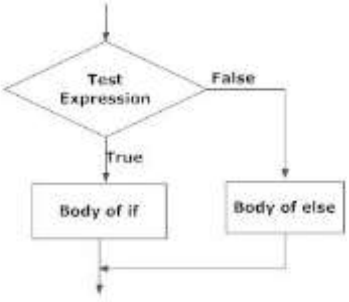


Fig: Operation of if...else statement

Example

```
num = 3
if num >= 0:
   print("Positive or Zero")
else:
   print("Negative number")
```

3.Explain if...elif...else Statement

The elif statement helps you to check **multiple expressions** and it **executes** the code as soon as one of the conditions evaluates to **True**.

Syntax:

```
if test expression:
   Body of if
elif test expression:
   Body of elif
else:
   Body of else
```

The elif is short for else if. It allows us to check for multiple expressions. If the condition for if is False, it checks the condition of the next elif block and so on. If all the conditions are False, the body of else is executed. The if block can have only one else block. But it can have multiple elif blocks.

Flowchart

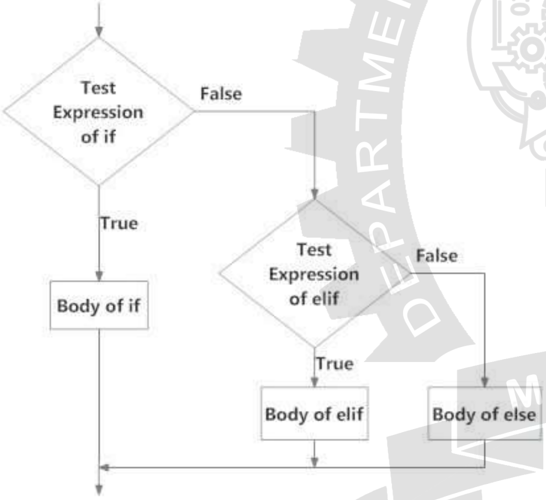


Fig: Operation of if...elif...else statement

Example

```
num=3
if num > 0:
   print("Positive number")
elif num == 0:
   print("Zero")
else:
   print("Negative number")
```

4.Explain nested if statements in python

We can have a if...elif...else statement inside another if...elif...else statement. This is called nesting in computer programming. Any number of these statements can be nested inside one another. Indentation is the only way to figure out the level of nesting.

Example

```
num = float(input("Enter a number: "))
if num >= 0:
   if num == 0:
       print("Zero")
   else:
       print("Positive number")
else:
   print("Negative number")
```

5.What is Looping Statements in python?

The flow of the programs written in any programming language is sequential by default. Sometimes we may need to alter the flow of the program. The execution of a specific code may need to be repeated several numbers of times.

For this purpose, The programming languages provide various types of loops which are capable of repeating some specific code several numbers of times.

In Python, Iteration (Loops) statements are as follows :-

1. While Loop
2. For Loop

Explain For Loop in Python

The for loop in Python is used to iterate over a sequence (list, tuple, string). Iterating over a sequence is called traversal.

Syntax:

```
for val in sequence:
   loop body
```

Here, val is the variable that takes the value of the item inside the sequence on each iteration. Loop continues until we reach the last item in the sequence. The body of for loop is separated from the rest of the code using indentation.

Flowchart

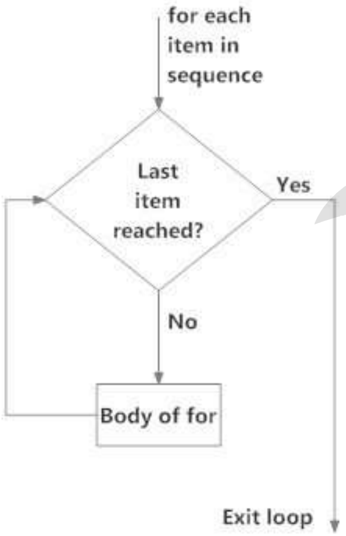


Fig: operation of for loop

Example

```
numbers = [6, 5, 3, 8, 4, 2, 5, 4, 11]
sum = 0
```

```
for val in numbers:
    sum = sum+val
print("The sum is", sum)
```

6.What is while loop in Python?

The while loop in Python is used to iterate over a block of code as long as the test expression (condition) is true. We generally use this loop when we don't know the number of times to iterate beforehand.

Syntax

```
while test_expression:
    Body of while
```

In the while loop, test expression is checked first. The body of the loop is entered only if the test_expression evaluates to True. After one iteration, the test expression is checked again. This process continues until the test_expression evaluates to False. In Python, the body of the while loop is determined through indentation.

Flow Chart

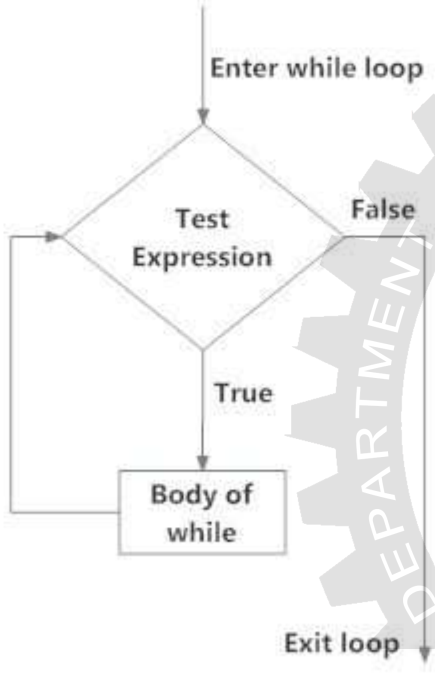


Fig: operation of while loop

Example

```
n = 10
sum = 0
i = 1

while i <= n:
    sum = sum + i
    i = i+1    # update counter

# print the sum
print("The sum is", sum)
```

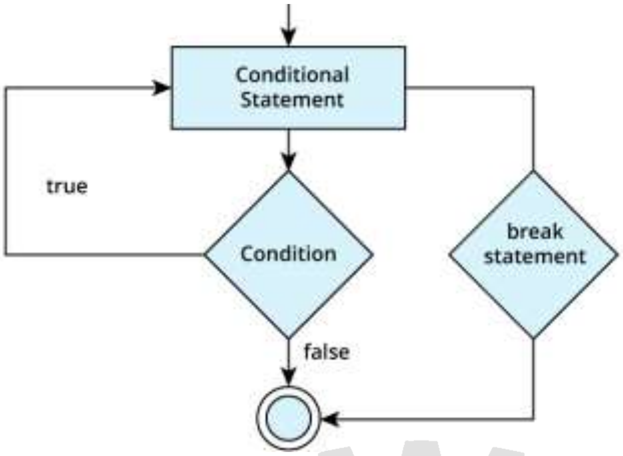
7.What are the different loop control statements in python?

Python supports 3 basic Loop Control Statement

Break Statement:

It terminates the current working loop and passes the control to the next statement, and if the break statement resides inside the nested loop, it passes control to the outer loop. It can be used with both while and for loops.

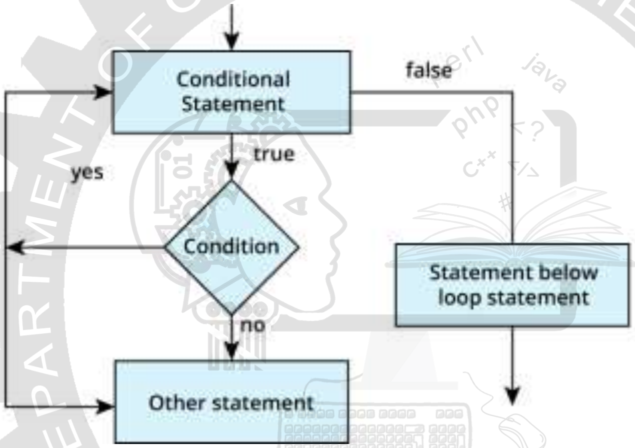
Flowchart



Continue Statements

The continue statement is used in the program to skip any iteration, such as when we don't want to print a certain statement. It skips the current iteration and executes the very next iteration in the loop.

Flowchart



Pass statement:

It works like a null operation because nothing happens when the pass statement is executed. It is used in Python when any statement is required syntactically, but we don't want any command or code lines to be executed.

Uses of Pass Statement:

- It can be used in Stubs.
- It can be used while writing empty loops.
- It can be used for empty control statements.
- It can also be used to implement empty functions and classes.

8.Explain range function?

The range() function is a built-in-function used in python, it is used to generate a sequence of numbers.

Syntax

The syntax of range() function is shown below:

range(start, stop, step)

***Start:** Optional — An integer number that specifies where to start (Default value is 0)*

Stop: Required — An integer number that specifies where to stop.

Step: Optional — An integer number that specifies how much to increment the number (Default value is 1)

Examples

```
x = range(10)
for i in x:
   print(i)
```

Output

```
0
1
2
3
4
5
6
7
8
9
```

```
x = range(1, 10)
for i in x:
   print(i)
```

output

```
1
2
3
4
5
6
7
8
9
```

```
x = range(1, 10, 2)
for i in x:
   print(i)1
3
5
7
9
```