Assignment 1

1.*Why do we call python as general purpose and a high level programming language?*

A. Python is an **object-oriented**, **high-level programming language**. Object-oriented means this language is based around **objects** (such as data) rather than functions, and high-level means it's easy for humans to understand.

Python is a **general-purpose language**, meaning it can be used to create a variety of different programs and isn't specialized for any specific problems. This versatility, along with its beginner-friendliness, has made it one of the most-used programming languages today.

2. *Why is Python called a dynamically typed language?*

A. Python Programming is widely used in Artificial Intelligence, Natural Language Generation, Neural Networks and other advanced fields of Computer Science .It is a powerful language that you can use to create games, write GUIs, and develop web applications. Due to wide applicability ,it is called a dynamic typed language.

3. . *List some pros and cons of Python programming language?*

A. **Pros of Python.**

- **Easy to learn** and understand.
- Python has a **vast standard library** that offers you nearly all the functions you may need for your task .
- Python **boosts productivity**; its simplicity allows developers to concentrate on solving problems rather than studying syntax of the language behavior.
- It is **interpreted language** ie. (Excute code one by one line). It is helpful in finding error in program. Since there is no compilation step, editing, debugging and testing is fast.
- **Portability**:-To run your program on different platforms, you don't need to modify the code.
- **Dynamically typed**:-Till you run the code, Python isn't aware of the variable type, and assigns the data type automatically while the code is being executed.
- Python is an **open-source language**, meaning it can be used and distributed freely; anyone can download the source code, make alterations, and distribute their own version of the language.
- It provide **rich data type** and easier to read syntax.

Cons of python

- **Poor memory efficiency** as Python needs a lot of memory space.
- **Slow speed a**s it executes the code one line at a time, the speed of execution often is hampered.
- **Weak in mobile computing:-**Python is used for backed programming; due its high memory usage and slow speed, it is generally not used for frontend programming or mobile app development.
- **Run time error**:-The data types of variables in Python can change suddenly. A variable holding a string may contain an integer later this can lead to runtime errors .

4. In what all domains can we use Python?

A. Python is commonly used for developing **websites and software, task automation, data analysis , and data visualization**. It is also used in domains such as **artificial intelligence, machine learning** and **deep learning** .

5. What are variable and how can we declare them?

A. A Python variable is a name given to a memory location. It is the basic unit of storage in a program . A variable is created the moment we first assign a value to it. Example:-

```
Name="Kaushal"
Age=30
Print(Name)
Print(Age)
Output: Kaushal
        30
```

6. How can we take an input from the user in Python?

A. **input ():** This function first takes the input from the user and converts it into a string. The type of the returned object always will be <type 'str'>. It does not evaluate the expression it just returns the complete statement as String. Example:-

```
#program showing input function
name = input("enter your name:")
Print(name)
Run:-
enter your name:
Output: Kaushal Singh
                Kaushal Singh
```

7. What is the default datatype of the value that has been taken as an input using input() function?

A. **input() function** is used to take user input. By default, it returns the user input in form of a string.

8. What is type casting?

A. Type Casting is the method to convert the variable data type into a certain data type in order to the operation required to be performed by users.

There can be two types of Type Casting in Python –

- Implicit Type Casting
- Explicit Type Casting

```
A=10 #int variable
Print(type(A))#implicit type casting
Output:<type 'int'>
```

explicit type casting

A=100 # int variable

n= float(A) # type casting into float

Print(type(n))

Output: <type 'float'>

9. Can we take more than one input from the user using single input() function? If yes, how? If no, why?

A. Python user can take multiple values or inputs in one line by two methods.

- Using split() method
- Using List comprehension

Using split() method :

This function helps in getting multiple inputs from users. It breaks the given input by the specified separator. If a separator is not provided then any white space is a separator .

Syntax:

Input().split(separator , maxsplit)

Program showing multiple input using split method

```
x, y, z =input ("enter the three values:-").split()
print ("Total number of students: ",x)
print (" Number of boys is: ", y)
print (" Number of girls is: ",z)
print ()
```

output:

enter the three value:- 50 20 30

Total number of students: 50

Number of boys is: 20

Number of girls is: 30

10. What are keywords?

A. Keywords are some **predefined and reserved words** in python that have special meanings. Keywords are used to define **the syntax** of the coding. The keyword cannot be used as an identifier, function, and variable name. example:- **int ,float , if, else, return, for, dict** etc.

All the keywords in python are written in lower case except **True , False** and **None.**

Q11. Can we use keywords as a variable? Support your answer with reason ?

A. Keywords can't be used as a variable. A keyword is any word that has special meaning to the interpreter (or compiler) of a programming language. These have also been known as reserved words in the past, which conveys a bit more meaning: they are reserved for the programming language and can't be used for anything else.

12. What is indentation? What's the use of indentaion in Python?

Python indentation is a way of telling a Python interpreter that the group of statements belongs to a particular block of code. A block is a combination of all these statements.

Most programming languages like C, C++, and Java use braces { } to define a block of code. Python uses indentation to highlight the blocks of code.

Whitespace is used for indentation in Python. All statements with the same distance to the right belong to the same block of code.

If a block has to be more deeply nested, it is simply indented further to the right.

example:-

```
a= 10
if  a>15:
    print("a is greater")
else:
    print("a is smaller ")
```

13. How can we throw some output in Python?

A. The **print() function** prints the specified message to the screen, or other standard output device. The message can be a string, or any other object, the object will be converted into a string before written to the screen.

14. What are operators in Python?

A. Python Operators in general are used to perform operations on values and variables. These are standard symbols used for the purpose of logical and arithmetic operations.

- **OPERATORS**: Are the special symbols. Eg- + , * , /, etc.
- **OPERAND**: It is the value on which the operator is applied.

Types of operators:-

- Airthmatic operator (+, - , % , * , // , /)
- Comparison operators (< , > , == , != ,>= , <= , is not, is)
- Logical operators (and , or , not)
- Bitwise operators (&, ^ , >> , <<)
- Membership operators (in , not in)

Q15. What is difference between / and // operators?

A. /' is the division operator and '//' is the floor division operator both belong to the arithmetic operators. '/' is used for the normal division of two numbers.

'//' is used to obtain the smallest integer nearest to the quotient obtained by dividing two numbers.

'Let us see an example to understand this.

a = 19

b = 4

print(a // b) #This prints output as 4

print(a / b) #This prints output as 4.75

So, if the quotient obtained by dividing two numbers is not an integer, then operators '/' and '//' will return different answers.

16. Write a code that gives following as an output.

iNeuroniNeuroniNeuroniNeuron

A. name= 'iNeuron'

*print (name*4)*

output: ***iNeuroniNeuroniNeuroniNeuron***

17. Write a code to take a number as an input from the user and check if the number is odd or even.

A. num=int(input("Enter the value :-"))

if num % 2 == 0:

print (" num is even")

else:

print ("num is odd")

output:-

enter the value :- 45

num is odd

18. What are boolean operator?

A. The logical operators **and, or** and **not** are also referred to as boolean operators. While and as well as or operator needs two operands, which may evaluate to true or false, not operator needs one operand evaluating to true or false.

Q19. What will the output of the following?

A. 1 or 0 =>> 1

B. 0 and 0 =>> 0

C. True and False and True =>> False

D. 1 or 0 or 0 =>> 1

E. or 0 or 0

20. What are conditional statements in Python?

A. Conditional statements are also known as decision-making statements. We need to use these conditional statements to execute the specific block of code if the given condition is true or false.

In Python , we can achieve decision making by using the following statements:

- if statements
- if-else statements
- elif statements
- Nested if and if-else statements
- elif ladder

Q21. What is use of 'if', 'elif' and 'else' keywords?

A. ***if statement:*** It decides whether certain statements need to be executed or not. It checks for a given condition, if the condition is true, then the set of code present inside the " if " block will be executed otherwise not.
Syntax: *if (expression== true):*

Block of codes

else:

Block of codes

elif statements: elif" statement is used to check multiple conditions only if the given condition is false. It's similar to an "if-else" statement and the only difference is that in "else" we will not check the condition but in "elif" we will check the condition.

Syntax: *if (conditions):*

elif(conditions):

else:

if else statements: *if statement* executes the code block when the condition is true. Similarly, the *else* statement works in conjuncture with the *if statement* to execute a code block when the defined *if condition* is false.

```
Syntax:

if condition:

        # execute code if condition is true

else:
     # execute code if condition if False
```

22. Write a code to take the age of person as an input and if age >= 18 display "I can vote". If age is < 18 display "I can't vote".

A. *age_person = int(input(" enter the age of the person:")*

if (age_person >= 18):

print ("I can vote")

else:

print("I can't vote")

output:-

enter the age of the person:***22***

I can vote

Q23. Write a code that displays the sum of all the even numbers from the given list. numbers = [12, 75, 150, 180, 145, 525, 50] .

sum=0

numbers = [12, 75, 150, 180, 145, 525, 50]

for i in numbers:

if i %2 == 0:

sum= sum+ i

print(" The sum of all even number in a list is:", sum)

output:-

The sum of all even number in a list is : ***392***

24. Write a code to take 3 numbers as an input from the user and display the greatest no as output.

A. program on next page

a = int(input ("enter the value:"))

b = int(input ("enter the value:"))

c = int(input ("enter the value:"))

if a > b and a > c:

print("greatest number is ", a)

```
if b > a and b> c :
    print(" greatest number is ", a)
else:
print("greatest number is ", c)
```

25. write a program to display only those number from the list which satisfy following conditions

- *the number must be divisible by 5*
- *if the number is greater than 150 then skip the number and move to next number*
- *if the number is greater than 500 then stop the loop*

```
numbers = [12, 75, 150, 180, 145, 525, 50]
list=[]
for i in numbers:
    if  i % 5 == 0:
        if i <= 150 :
         list.append(i)
          if  i > 500:
              break
print(list)
```

output:

[75,150,45]