PRACTICAL PROGRAMS

Python Basic

PROGRAM NO-1:

AIM- Write a Python program to print the following string in a specific format.

SOURCE CODE-

```
Twinkle, twinkle, little star,
        How I wonder what you are!
                Up above the world so high,
                Like a diamond in the sky.
Twinkle, twinkle, little star,
        How I wonder what you are!
```

```
print("Twinkle, twinkle, little star, \n\tHow I wonder what you are!
\n\t\tUp above the world so high, \n\t\tLike a diamond in the sky.
\nTwinkle, twinkle,little star, \n\tHow I wonder what you are!")
```

OUTPUT-

```
Twinkle, twinkle, little star,
        How I wonder what you are!
                Up above the world so high,
                Like a diamond in the sky.
Twinkle, twinkle, little star,
        How I wonder what you are!
```

RESULT- The source code is executed successfully and the output was verified.

PROGRAM NO-2

AIM- Write a Python program to print 5 lines about yourself using print() function.

SOURCE CODE

OUTPUT-

```
#Project - 1 Write 5 lines about your self
a = input("Write your name:")
b = int(input("Write your class:"))
c = int(input("Write your roll no.:"))
d= input("Write your father's name:")
e = input("Write your hobbies:")
print("Hello World")
print("1. My name is ",a)
print("2. I'm studying in class ",b)
print("4. My roll no. is ",c)
print("5. My Father's name is ",d)
print("6. My hobbies are ",e)
```

RESULT- The source code is executed successfully and the output was verified.

PROGRAM NO-3

AIM- Write a program to make a simple calculator.

SOURCE CODE-

```
# Python program for simple calculator

# Function to add two numbers
def add(num1, num2):
    return num1 + num2
# Function to subtract two numbers
def subtract(num1, num2):
    return num1 - num2

# Function to multiply two numbers
def multiply(num1, num2):
    return num1 * num2
# Function to divide two numbers
# Take input from the user
select = int(input("Select operations form 1, 2, 3, 4 :"))

number_1 = int(input("Enter first number: "))
number_2 = int(input("Enter second number: "))

if select == 1:
    print(number_1, "+", number_2, "=",
                                add(number_1, number_2))
```

```
elif select == 2:
        print(number_1, "-", number_2, "=",
                                        subtract(number_1, number_2))
elif select == 3:
        print(number_1, "*", number_2, "=",
                                        multiply(number_1, number_2))
elif select == 4:
        print(number_1, "/", number_2, "=",
                                        divide(number_1, number_2))

else:
        print("Invalid input")
```

OUTPUT-

RESULT- The source code is executed successfully and the output was verified

```
Please select operation -
1. Add
2. Subtract
3. Multiply
4. Divide

Select operations form 1, 2, 3, 4 :4
Enter first number: 21
Enter second number: 25
21 / 25 = 0.84
>>>
```

Multiple-Choice Questions

1. Find the invalid variable among the following:

a. 1st_string

b. my_string_1

c. _

d. foo

Answer: (a) 1st_string

2. The order of precedence in the Python language is:

A) Exponential

B) Parentheses

C) Division

D) Multiplication

E) Subtraction

F) Addition

a. B,A,D,C,F,E

b. A,B,D,C,F,E

c. A,B,C,D,E,F

d. B,A,D,C,E,F

Answer: (a) B,A,D,C,F,E

3. Which one of these is incorrect?

a. float(‘nan’)

b. float(‘inf’)

c. float(’12+34′)

d. float(’56’+’78’)

Answer: (c) float(’12+34′)

4. The value of the Python expression given below would be:

4+2**5//10

a. 77

b. 0

c. 3

d. 7

Answer: (d) 7

5. The return value for trunc() would be:

a. bool

b. float

c. int

d. None

Answer: (c) int

6. What is the output of the Python code given below, if the date of the system is June 21st, 2017 (Wednesday)?

[] or {}

{} or []

a. [ ] [ ]

b.[ ] { }

c.{ } { }

d.{ }[ ]

Answer: (d){} []

7. The output of this Python code would be:

s='{0}, {1}, and {2}’

s.format(‘hi’, ‘great’, ‘day’)

a. ‘hi, great, and day’

b. ‘hi great and day’

c. ‘hi, great, day’

d. Error

Answer: (a) ‘hi, great, and day’

8. The output of this Python code would be:

```
a = [‘mn’, ‘op’]

for i in a:
    i.upper()
      print(a)
```

a. [None, None]
b. [‘MN’, ‘OP’]
c. [‘mn’, ‘op’]
d. None of the above

Answer: (c) [‘mn’, ‘op’]

9. The output of this Python code would be:

print(“mno. PQR”.capitalize())
a. Mno. Pqr
b. Mno. pqr
c. MNO. PQR
d. mno. pqr

Answer: (b) Mno. pqr

10. Which arithmetic operators can we NOT use with strings?
a. –
b. +
c. *

d. All of the above

Answer: (a) –

11. Which function do we use to shuffle a list(say list1)?
a. shuffle(list1)
b. list1.shuffle()
c. random.shuffleList(list1)
d. random.shuffle(list1)

Answer: (d) random.shuffle(list1)

12. In the following statements of Python, which ones will result into the output: 6?

A = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
a. A[3][2]
b. A[2][3]
c. A[1][2]
d. A[2][1]

Answer: (c) A[1][2]

13. Is this code valid in Python?

>>> m=6,7,8,9
>>> m
a. No, many values will unpack
b. Yes, (6,7,8,9) will be printed
c. Yes, 6 will be printed
d. Yes, [6,7,8,9] will be printed

Answer: (b) Yes, (6,7,8,9) will be printed

14. Which function removes a set’s first and the last element from a list?
a. pop
b. remove
c. dispose
d. discard

Answer: (a) pop

15. The output of this Python code would be:
>>> x={1:”X”,2:”Y”,3:”Z”}
>>> del x
a. the del method does not exist for dictionary
b. the del would delete the values present in dictionary
c. the del would delete the entire dictionary

d. the del would delete all the keys in dictionary

Answer: (d) the del would delete all the keys in dictionary

16. The output of this Python code would be:

```
sum(1,2,3)
sum([2,4,6])
```

a. 6, 12

b. Error, Error

c. Error, 12

d. 6, Error

Answer: (c) Error, 12

17. The output of this Python code would be:

```
def find(x, **y):
        print(type(y))
        find('letters',X='1′,Y='2′)
```

a. Dictionary

b. An exception is thrown

c. String

d. Tuple

Answer: (a) Dictionary

18. Which one of these is NOT true about recursion?

a. We can replace a recursive function by a non-recursive function

b. The memory space taken by the recursive functions is more than that of non-recursive function

c. Running a recursive function is faster as compared to a non-recursive function

d. The process of recursion makes it easier for users to understand a program

Answer: (c) Running a recursive function is faster as compared to a non-recursive function

19. The output of this Python code would be:

```
a = ['mn', 'op']
print(len(list(map(list, a))))
```

a. 4

b. 2

c. Not specified

d. Error

Answer: (d) Error

SHORT AND LONG QUESTION ANSWER:

1. What are some key features of Python?

Ans: Easy to code, free, open-source, interpreted, high-level and integrated, object-oriented, portable, dynamically typed, extensible, and offers GUI programming support.

2. What are keywords in Python?

Ans: Reserved and predefined words with special meanings used to define the coding syntax are called keywords in Python. Except for True and False, we always write these keywords in Python in lowercase. They can also not be used as variable, function, or identifier names.

3. What are the advantages of Python?

Ans: Python has English-like syntax, is concise, object-oriented, high-level, interpreted, free, and open source. Hence, Python is easy to learn, access, and write code in. Also, Python code is more readable, portable, and maintainable, and Python offers vast library support. No wonder then that Python improves the productivity of software developers and makes the process of software development smoother.

4. How many data types are there in Python?

Ans: Python has five standard data types:

String, Numbers, Tuple, List, Dictionary

5. What is the use of Jupyter Notebook?

Ans: The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations, and narrative text. Uses include data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more.

6. What do you need virtual environment?

Ans: A virtual environment is a tool that helps to keep dependencies required by different projects separate by creating isolated python virtual environments for them. This is one of the most important tools that most Python developers use.

7. Explain the python list.

Ans. A Python list is a sequence of comma separated items, enclosed in square brackets [ ]. The items in a Python list need not be of the same data type. In Python, a list is a sequence data type. It is an ordered collection of items.

8. What is standared data types of Python?

Ans: Python has five standard data types – Numbers, String, List , Tuple, Dictionary

9. Define the python tuple.

Ans: Python Tuple is a collection of objects separated by commas. In some ways, a tuple is similar to a Python list in terms of indexing, nested objects, and repetition but the main difference between both is Python tuple is immutable, unlike the Python list which is mutable.

T=(1, “Manoj Kumar”)

10. **Explain the Decision making statements of python.**

Ans: Decision making statements can direct a program to execute on the based of instructions if a certain condition is True or another if it's False. Different decision making structures can be used such as if statements, else-if statements, and nested if else statements for decision making in python programming language.

Python's decision making functionality is in its keywords − **if..elif...else**. The if keyword requires a boolean expression, followed by colon symbol.

The colon (:) symbol starts an indented block. The statements with the same level of indentation are executed if the boolean expression in if statement is **True**. If the expression is not True (False), the interpreter bypasses the indented block and proceeds to execute statements at earlier indentation level.

Python − The if Statement

```
discount = 0
amount = 1200

# Check he amount value
if amount > 1000:
  discount = amount * 10 / 100

print("amount = ", amount - discount)
```

11. Define the Operators in Python.

Ans: In Python programming, 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. In this article, we will look into different types of **Python operators.**

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

Types of Operators in Python

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

12. Differentiate between For loop and While Loop.

Ans: For Loop

A **for loop** is a control flow statement that executes code for a predefined number of iterations. The keyword used in this control flow statement is "**for**". When the number of iterations is already known, the *for* loop is used.

The *for* loop is divided into two parts −

Header − This part specifies the iteration of the loop. In the header part, the loop variable is also declared, which tells the body which iteration is being executed.

Body − The body part contains the statement executed per iteration.

- The initialization, condition checking, and iteration statements are written at the beginning of the loop.
- It is used only when the number of iterations is known beforehand.
- If the condition is not mentioned in the 'for' loop, then the loop iterates the infinite number of times.
- The initialization is done only once, and it is never repeated.
- The iteration statement is written at the beginning.
- Hence, it executes once all statements in the loop have been executed.

Syntax

```
for(initialization; condition; iteration){
  //body of the 'for' loop
}
```

Example:

```
# input list
inputList = [10, 20, 30, 40, 50]
print("Input list elements:")
# traversing through all elements of the list using for loop
for element in inputList:
  # printing each element of the list
  print(element)
```

While Loop

A loop that runs a single statement or a set of statements for a given *true* condition. This loop is represented by the keyword "**while**". When the number of iterations is unknown, a **while loop** is used. The statement is repeated until the Boolean value is *false*. Because the condition is tested at the beginning of a *while* loop, it is also known as the pre-test loop.

- The initialization and condition checking are done at the beginning of the loop.
- It is used only when the number of iterations isn't known.
- If the condition is not mentioned in the 'while' loop, it results in a compilation error.
- If the initialization is done when the condition is being checked, then initialization occurs every time the loop is iterated through.
- The iteration statement can be written within any point inside the loop.

Syntax

```
while ( condition) {
  statements;
  //body of the loop
}
```

Example

```
# Initializing a dummy variable with 1
i = 1
# Iterate until the given condition is true
while i < 10:
  # printing the current value of the above variable
  print(i)
  # incrementing the value of i value by 1
  i += 1
```

13. What is python Identifiers? **Rules for Naming Python Identifiers**

Ans: Identifier is a user-defined name given to a variable, function, class, module, etc. The identifier is a combination of character digits and an underscore. They are case-sensitive i.e., 'num' and 'Num' and 'NUM' are three different identifiers in python. It is a good programming practice to give meaningful names to identifiers to make the code understandable.

Rules for Naming Python Identifiers

- It cannot be a reserved python keyword.
- It should not contain white space.
- It can be a combination of A-Z, a-z, 0-9, or underscore.
- It should start with an alphabet character or an underscore (_).
- It should not contain any special character other than an underscore (_).

14. Write the program to find simple interest using Jupiter notebook.

```
P = int(input("Enter the principal amount :"))
T = int(input("Enter the time period :"))
R = int(input("Enter the rate of interest :"))
  si = (p * t * r)/100
  print('The Simple Interest is', si)
```