Class IX
Python Programming Language

Theoretical Questions

1. What is Python?

   Python is a high-level, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation.

2. In which area Python Programming is useful for?

   Python Programming is useful in the area of:

   a. web development (server-side),
   b. software development,
   c. mathematics,
   d. system scripting.

3. What can Python do?

   Python can be used on a server to create web applications.

   a. Python can be used alongside software to create workflows.
   b. Python can connect to database systems. It can also read and modify files.
   c. Python can be used to handle big data and perform complex mathematics.
   d. Python can be used for rapid prototyping, or for production-ready software development.

4. Why Python?

   Python is preferred programming language due to the following reasons:

   a. Easy to learn: It has fewer keywords and simple structure of instructions, so less efforts to learn Python.
   b. Free and open source
   c. Portable: A Python program can run on wide variety of hardware platforms.
   d. Extensive library support: Python saves times and effort using the support of extensive library.

   As an example, YouTube has been largely written in Python. Google makes extensive use of Python. NASA uses Python for scientific programming tasks.

5. What are the two modes of Python?

   The two modes of Python are interactive and script mode.

6. What is interactive mode. List it's advantages and drawbacks.

   As the name suggests, Interactive mode allows us to interact with OS. When we type Python statement, interpreter displays the result(s) immediately.

   Advantages:

   - Python, in interactive mode, is good enough to learn, experiment or explore.
   - Working in interactive mode is convenient for beginners and for testing small pieces of code.

   Drawback:

   - We cannot save the statements and have to retype all the statements once again to re-run them.

7. What is script mode. List its advantages and drawbacks.

   In script mode, we type python program in a file and then use interpreter to execute the content of the file. Scripts can be saved to disk for future use. Python scripts have the extension .py, meaning that the filename ends with .py.

8. What are variables and how to create the same in Python?
Variables – Variables are containers for storing data values.
Creating Variables – Python has no command for declaring a variable.

A variable is created the moment you first assign a value to it. Variables do not need to be declared with any type and can even change type after they have been set.

For example:
x = 5
y = "John"

Here x is variable which is assigned a value 5 which is automatically declared as integer as data type and y is variable which is assigned a value ‘John’ which is automatically declared as string data type.

9. Discuss Data Types is Python?

Python supports a number of data types such as number, string, list, tuple, set and dictionary. We will discuss about number and string data type here:

Number: The following are the two important number data type:

a. int – int stands for integer which is a positive or a negative number without a decimal point. There are three types of integers in Python:
Example: a = 7; b=-90
b. float – float or floating-point number is positive or negative number with a fractional part.
Example: c = 9.7; d = -88.9

Strings: Strings in Python are a sequence of characters. These characters are enclosed within single quotes (‘), double quotes (“) or triple quotes (“). Triple quotes are used when the string value extends over more than one line.

Example of strings:
name = “Myra”
section = ‘A’
message = ‘‘‘A dream does not become reality through magic; it takes sweat, determination, and hard work. ~ Colin Powell’’’

10. What are operators in Python?
Operators are special symbols that are used to perform operations on variables and values. Operators are used in everyday Mathematics, like +, -, x, ÷ to perform addition, subtraction, multiplication, and division of values. Python also has a number of operators that help us perform operations of various types.

11. What is assignment operator?
Assignment operator (=) is operator which is used to assign a value to a variable.
For example:
x = 5
y = "John"

12. What are arithmetic operators in Python? Explain with an example.
Arithmetic operators are operators used to perform mathematical operation on variables and value.
For example, in 5 + 9, ‘+’ is arithmetic operator performing addition on values 5 and 9.
Similarly, in p / q, ‘/’ is arithmetic operator performing division on p and q.

13. List all basic types of arithmetic operators used in Python. Use examples.

The following are the list of arithmetic operators most used in Python:

Operator	Description	Example	Result	Comment
+	Addition	4 + 3	7	
-	Subtraction	9 - 18	-9	
*	Multiplication	7 * 5	35	
/	Division	36 / 6	6	
//	Floor Division	7.2 // 2	3.0	divides the truncates the fractional part from the result.
%	Remainder	35 % 6	5	divides the two operands and gives the remainder resulting.
**	Exponent	3 ** 4	81	returns base raised to power exponent. 3^4 here.

14. What are relational operators in Python? Explain with an example.

Relational operators are known as comparison operators as they are used to compare two values. When we use relational operators, the output is either true or false.

For example, p = 20, q = 25, so p == q is false and p != q is true as 20 is not equal to 25.

15. List all basic types of relational operators used in Python. Use examples.

The following are the list of arithmetic operators most used in Python:

Operator	Description	Examples

==	Equal to (Equality)	a = 10, b = 8 so, a == b is false
		x = 15, y = 15, so x == y is true

!=	Not equal to (Inequality)	a = 10, b = 8 so, a != b is true
		x = 15, y = 15 so x != y is false

<	Less than	a = 10, b = 8 so, a < b is false
		x = 15, y = 15, so x < y is false

<=	Less than or equal to	a = 10, b = 8 so, a <= b is false
		x = 15, y = 15, so x <= y is true

>	Greater than	a = 10, b = 8 so, a > b is true
		x = 15, y = 15, so x > y is false

>=	Greater than or equal to	a = 10, b = 8 so, a >= b is true
		x = 15, y = 15, so x >= y is true

16. Can relational operators be used with String values? Explain with examples.

Equality (==) and Inequality (!=) operators can be used to compare the values of the string.
For example,

a. x = “Hi” and y = “Hello”, so x == y is false as Hi and Hello are not same and x != y is true.
b. a = “Good Bye” and b = “Good Bye”, so a == b is true and a != b is false.

17. What are logical operators in Python? List all logical operators. Explain with an example.

Logical operators are used when multiple conditions are to be tested. They give the result as true or false.
The following are the list of logical operators:

Operators	Description
`and`	The **and** operator gives a True value only if both conditions are True. It gives a False value if either or both of the two conditions are False.
`or`	The **or** operator gives a True value only if both or either of the two conditions is True. It gives a False value only if both the conditions are False.
`not`	The **not** operator gives a True value if the condition is False and a False value if the condition is True.

For example, assume age = 12, sports = “cricket”

The table below shows the result of logical operators when used with variables age and sports.

Expression	Result
`age > 10 and sports == “football”`	`False`
`age > 10 or sports == “football”`	`True`
`not (sports == “cricket”)`	`False`

18. What is conditional statement?
A statement that helps us check conditions and choose a set of statements to be executed.

19. What is if… else statement? What is the syntax?
If… else statement is a conditional statement used when we want the program to choose between two blocks of code depending on when the condition is true or false.

Syntax:

```
if condition:
   statement (s) {block of statement(s) executed when condition is true}
else:
   statement (s) {block of statement(s) executed when condition is false}
```

20. What is `if… elif… else` statement? What is the syntax? Use examples wherever necessary.
`if… elif… else` is similar to `if… else` statement and used in situation where multiple conditions are to be checked.

For example, grading system of school examination such as A, B, C, D, E. different marks scored by candidates result in different grades, so multiple conditions are to be checked.

Syntax:

```
if condition:
   statement (s) {block of statement(s) executed when if condition is true}
elif:
   statement (s) {block of statement(s) executed when elif condition is true}
else:
   statement (s) {block of statement(s) executed when none of the condition is true}
```

21. What is a loop?
While programming we are often required to repeat a set of statements multiple times. Repetition of a set of statements in a program is done using loops which are block of code executed multiple times. In Python, we can create loop using for and while statements.

22. What is `range ( )` FUNCTION in Python? What is its syntax? Explain with examples.
The built in function range ( ) in Python is used to generate a sequence of numbers. This function can take a maximum of three integer arguments.

Syntax:
`range ([start], stop, [step])`

Argument	Description
`start`	Optional Argument: An integer specifying the starting value of the sequence. Default value is 0.

`stop`	Required Argument: An integer specifying the value at which the sequence will stop. The stop value is not included in the sequence.
`step`	Optional Argument: An integer specifying the incrementation. Default value is 1.

Examples:

Example	**Read same as below**	**Sequence generated**	**Explanation**

`range (10, 21, 2)`	`range (10, 21, 2)`	`10, 12, 14, 16, 18, 20`	`Start value: 10`
			`Stop value: 21`
			`Step value: 2`

`range (1, 10)`	`range (1, 10, 1)`	`1, 2, 3, 4, 5, 6, 7, 8, 9`	`Start value: 1`
			`Stop value: 10`
			`Step value: 1 (default)`

`range (5)`	`range (0, 5, 1)`	`0, 1, 2, 3, 4`	`Start value: 0 (default)`
			`Stop value: 5`
			`Step value: 1 (default)`

`range (10, 0, -3)`	`range (10, 0, -3)`	`10, 7, 4, 1`	`Start value: 10`
			`Stop value: 0`
			`Step value: -3`

23. Discuss **for** LOOP in brief with syntax and example(s).

The for LOOP is used to iterate over a range of values or a sequence.

Syntax:

```
for <control-variable> in <sequence/item in range>:
   body of the loop {block of statement(s) to be executed}
```

Example: Write any of the program from Practical Questions Q. No. 9, 10 and 11

24. Discuss **while** LOOP in brief with syntax and example(s).

The while LOOP statement executes a block of code repeatedly as long as the test conditions is true.

Syntax:

```
while condition:
   body of the loop {block of statement(s) to be executed}
```

Example: Write any of the program from Practical Questions Q. No. 12 and 13

Fill in the blanks

1. _________ are special symbols that are used to perform operations on variables and values.
2. _________ is the development environment that helps us create, run and debug Python programs from a single interface.
3. == is a _________ operator.
4. _________ is an example of logical operator.
5. The != operator helps us check for _________.
6. If… else is a _________ statement.
7. We use _________ statement to check for multiple conditions.

Answers:

1. Operator
2. IDLE
3. relational
4. or
5. inequality
6. conditional
7. if… elif… else

Multiple Choice Questions

1. Which of the following statements is used to display the result of the expression 4^3.

```
a. print (3**4)     b. print (4%3)      c. print (4**3)     d. print (4*3)
```

2. Find the odd one out.

a. ==   b. >=   c. =   d. <=

3. For which of the following options will the condition (`fruit==‟apple″ or qty>10`) evaluate to `FALSE`?

```
a. fruit=‟mango″, qty=15                b. fruit=‟apple″, qty=5
c. fruit=‟apple″, qty=100               d. fruit=‟mango″, qty=5
```

4. For which of the following options will the condition `(fruit==‟apple″ and qty>10)` evaluate to `TRUE`?

```
a. fruit=‟mango″, qty=15                b. fruit=‟apple″, qty=5
c. fruit=‟apple″, qty=100               d. fruit=‟mango″, qty=5
```

5. What output will be displayed if score entered by the user is 15?

```
if score == 100:
   print ("You win")
elif score >= 50:
   print ("You did well")
elif score >= 25:
   print ("You should practice more")
else:
   print ("You do not know the game")
```

a. You win
b. You did well
c. You should practice more
d. You do not know the game

6. Which of the following statements help us set up a loop in Python?

a. for statement
b. while statement
c. do while statement
d. Both (a) and (b)

7. Which of the following is false regarding loops in Python?

   a. Loops are used to perform certain tasks repeatedly.
   b. The while loop is used when multiple statements are to be executed repeatedly until the given condition becomes False.
   c. The while loop is used when multiple statements are to be executed repeatedly until the given condition becomes True.
   d. The for Loop can be used to iterate over a range of values or a sequence.

8. How many times will the following loop run?

```
for i in range (1):
   print (i)
```

a. 0   b. 1   c. 2   d. 3

9. Which of the following loops will display numbers from 1 to 100?

```
a. for i in range (101):                b. for i in range (1, 100):
      print (i)                               print (i)
c. for i in range (100):                d. for i in range (1, 101):
      print (i)                               print (i)
```

10. How many times will the following loop run?

```
while (3==3):
   print ("Hello World")
```

a. does not run b. 2 times c. 3 times d. Infinite times

11. What series of numbers is generated if we use range (1, 5, 0)?

a. 1, 1, 1, 1, 1 b. 1, 2, 3, 4 c. 1, 2, 3, 4, 5 d. Error

Answers:

1. c	2. c	3. d	4. c	5. d	6. d
7. c	8. b	9. d	10. d	11. d	

Explanation:

1. ** is the operator used for exponent.
2. = is assignment operator while all others are relational or comparison operator.
3. Using 'or' operator, both conditions to be false for the entire statement to be false.
4. Using 'and' operator, both conditions to be true for the entire statement to be true.
5. Since score = 15, is neither = 100 nor >= 50 nor >=25, so statement after else will be executed.
6. for statement and while statement are used for loops in Python. There is no do while statement.
7. Self-explanatory
8. range (1) = range (0, 1, 1). So, the sequence generated is 0 (once). So, loop will run once.
9. range (1, 101) = range (1, 101, 1), so sequence generated is 1, 2, 3, … … …, 99, 100.
10. Since 3 will always be equal to 3, so loop will execute infinite time as the condition will never be false.
11. Step value in range ( ) function cannot be set to zero.

Practical Questions

1. Find the output of the following if the variables subject, fee and num have been assigned the following value:
subject = "English", fee = 2000, num = 100

Expression	Result
`subject != "Hindi"`	
`(fee + 100) <= 3000`	
`(num * 2) == (fee / 10)`	
`(subject == "English") and (fee == 2000)`	
`(num == 200) or (fee != 1000)`	

Answer:
As given in question, subject = "English", fee = 2000, num = 100

Expression	Result	Explanation
`subject != "Hindi"`	True	As the value of subject is assigned English in the question. So, subject is not equal to Hindi.
`(fee + 100) <= 3000`	True	fee + 100 = 2000 + 100 = 2100. So as 2100 <= 3000, result is True
`(num * 2) == (fee / 10)`	True	num * 2 = 100 * 2 = 200 fee / 10 = 2000 / 10 = 200 So as 200 == 200, result is True
`(subject == "English") and (fee == 2000)`	True	As subject is English and fee is also equal to 2000, so both conditions are true, so result is true.

`(num == 200) or (fee != 1000)`	`True`	The value of num given in the question is 100, so (num == 200) is false. The value of fee given in the question is 2000, so (fee != 1000) is true. As one of the conditions is true using 'or' operator, so answer is true.

2. Write a program that takes the name of the user and then displays a welcome message to the user. The welcome message should be:
<User name>, Welcome to the world of Python Programming

Program:
```
name = input ("Enter your name: ")
msg = name + ", Welcome to the world of Python Programming"
print (msg)
```

3. Write a program to take name and country from the user and then display the following message to the user:
Welcome <user name>
Your country <country name> is a beautiful country

Program:
```
u_name = input ("Enter your name: ")
c_name = input ("Enter name of your country: ")
line1 = "Welcome " + u_name
line2 = "Your country " + c_name + " is a beautiful country"
print (line1)
print (line2)
```

4. Write a program to take three integers from user and display their sum.

Program:
```
num1 = int (input ("Enter the first number: "))
num2 = int (input ("Enter the second number: "))
num3 = int (input ("Enter the third number: "))
sum = num1 + num2 + num3
print ("The sum of three numbers = ", sum)
```

5. Write a program to take length and breadth of a rectangle from the user and display its area and perimeter.

Program:
```
l = int (input ("Enter the length of the rectangle: "))
b = int (input ("Enter the breadth of the rectangle:"))

area = l * b
perimeter = 2 * (l + b)

print ("Area of the rectangle = ", area)
print ("Perimeter of the rectangle = ", perimeter)
```

6. Write a program that displays a message based on the age entered by the user.

Condition	Message
>= 18	You have the right to cast vote
< 18	You cannot cast a vote

Program:

```
age = int (input (“Enter your age: ”))
if age >= 18:
   print (“You have the right to cast vote”)
else:
   print (“You cannot cast a vote”)
```

7. Write a program to check the answer entered by the user.

Program:

```
print (“Answer by typing in capital letters”)
city = input (“Name the city in which Taj Mahal is located: ”)
if city == “AGRA”:
   print (“You guessed it right”)
else:
   print (“Wrong Answer”)
```

8. Write a program to decide the grade scored by candidate in an examination on the basis of marks scored out of 100 as per the following:

Marks Scored	Grade
90 and above	A
Marks >= 80 and less than 90 (80-90)	B
Marks >= 70 and less than 80 (70-80)	C
Marks >= 60 and less than 70 (60-70)	D
Mark less than 60	E

Program:

```
marks = int (input("Enter you marks out of 100: "))
if marks >= 90:
    print ("You have got A grade")
elif marks >= 80:
    print ("You have got B grade")
elif marks >= 70:
    print ("You have got C grade")
elif marks >= 60:
    print ("You have got B grade")
else:
    print ("You have got E grade")
```

9. Write a program to display the square of first 5 natural numbers.

Program:

```
n = 5
for i in range (1, n+1):
   print (i*i)
```

10. Write a program to display the table of a number.

Program:

```
num = int (input (“Enter the number whose table is to displayed: ”))
for i in range (1, 11):
   print (num, “*”, i, “=”, num*i)
```

11. Write a program to display the sum of even numbers between 1 and 100.

Program:

```
sum = 0
for i in range (2, 101, 2):
   sum = sum + i
print (“Sum of even number between 1 and 100: ”, sum)
```

12. Write a program to display odd numbers less than or equal to 15.

Program:

```
num = 1
while num <= 15:
   print (num)
   num = num + 2
```

13. Write a program to take in as many numbers as user wants and display their sum.

Program:

```
sum = 0
choice = ‘y’
while choice == ‘y’ or choice == ‘Y’:
   num = int (input (“Enter a number: ”))
   sum = sum + num
   choice = input (“Enter y to continue: ”)
print (“Sum of numbers entered by the user: ”, sum)
```

14. Rewrite the following statements after removing the errors.

```
a. Name = “Sheela’
b. print (sum is, 3+5)
c. p=input(float(“Enter a number”))
d. print {“I like”, “Swimming”)
e. age = int (input (Enter your age))
f. print (“it”; “is”; “a beautiful day”)
```

Answers: The following are the correct statements after removing errors:

```
a. Name = “Sheela”
b. print (“sum is”, 3+5)
c. p=float(input(“Enter a number”))
d. print ("I like", "Swimming")
e. age = int (input (“Enter your age”))
f. print ("it", "is", "a beautiful day")
```

15. Predict the output of the following code segments in Python.

```
1. print (“Output”)         2. m = 10+2                3. a=’12’
   a = 18                      n = m+5                    b=’34’
   b = 2                       print (“RESULT”)           print (a+b)
   print (“answer=”,a/b)       print (m)
                               print (n)

4. a = 3                    5. a = “Important”
   b = 2                       b = “Message”
   c = a * b                   print (b+“ is ”+a)
   d = c * b
   print (a, b, c, d)
```

Answers: The following will be the output of each of the code:

```
1. Output                 2. RESULT                 3. 1234
   answer= 9.0               12
                             17

4. 3 2 6 12               5. Message is Important
```

16. Consider the following code and answer the questions that follows:

```
name = input ("Enter you name: ")
n = int (input ("Enter number of saplings per row: "))
r = int (input ("Enter number of rows in the field: "))

print (name, "You have to plant", n*r, "saplings")
```

   a. How many variables used in the code?
   b. Which variable (s) refers to the value of type string?
   c. If the user enters the name as "Rishi", number of saplings per row as 10 and number of rows as 30 what will be the output of the code?

Answer:

   a. Three
   b. Variable 'name'
   c. The following will be the output:

```
Rishi, You have to plant 300 saplings
```

17. Consider the following code and answer the questions that follows:

```
marks = int (input ("Enter marks: "))
if marks >= 90:
   grade = "A"
elif marks >= 80:
   grade = "B"
elif marks >= 70:
   grade = "C"
else:
   grade = "D"
print ("Your grade is: ", grade)
```

   d. What is the data type of the value in the variable marks and grade?
   e. What grade will be assigned to a student if the marks are 76?
   f. Under what condition would a student be assigned the grade "D"?

Answer:

   a. Data type of marks is integer and data type of grade is string.
   b. Grade C will be assigned.
   c. Grade D will be assigned to student if the student score less than 70 marks.

18. Ten families reside in a society. Consider the given code and answer the questions that follow:

```
count_car = 0
for i in range (1, 11):
   print ("Resident Number: ", i)
   choice_car = input ("Enter Y if you own a car: ")
   if choice_car == "Y":
         count_car = count_car + 1
print ("Number of car owners: ", count_car)
```

a. What does the above program do?
b. How many times will the loop run?
c. What happens if the resident enters ‘y’ (alphabet y in lowercase) in response to the question?
d. What statements should be added to the program to display the count of residents who do not own a car?

Answer:

a. The above program will ask the resident individually if they own a car. Based on the response, the program will display the number of residents in society who owns a car.
b. 10 times.
c. The users will not be included in the resident who owns a car.
d. The following statement to be added at the end of the code:
```
print ("Number of residents without a car: ", 10 - count_car)
```

In the end, point to remember while write Python Program

1. Python is case sensitive. So, name and Name are not the same.
2. All the Python commands and keywords use lower case (small letters)
   Print (“My School is best”) is incorrect
   print (“My School is best”) is the correct usage of the Python print command
3. Indentation: Indentation refers to the spaces at the beginning of a code line.
   Where in other programming languages the indentation in code is for readability only, the indentation in Python is very important. Python uses indentation to indicate a block of code.
   A wrong indentation may lead to an error, or the program will return a different output other than as intended.

   For example: observe below code:
   ```
   Line 1:  if marks >= 90:
   Line 2:        grade = “A”
   ```

   In the above code, notice the space in the line 2, that is, grade 2 start a little away from the actual start as it is the part which is to be executed if the condition is true.

4. Use colon (:) correctly in case of if… else… statement, for loop and while loop.
5. Use brackets correctly after the command. Mostly “(” and “)” to be used correctly.
6. Use single, double, and triple quotes correctly.