SKV VIDHYAASHRAM SR. SEC. SCHOOL
KANDAMPALAYAM, TIRUCHENGODE

CBSE - GR-XII - COMPUTER SCIENCE WITH PYTHON
(2021 - 2022)

TERM- 1 BOARD EXAMINATIONS

UNIT-1 PYTHON MCQ'S
(CHAPTER WISE AND TOPIC WISE)

CHAPTER-1 - PYTHON REVISION TOUR-1
Questions and Answers – Variable Names

1. Is Python case sensitive when dealing with identifiers?
a) yes
b) no
c) machine dependent
d) none of the mentioned

Answer: a
Explanation: Case is always significant.

2. What is the maximum possible length of an identifier?
a) 31 characters
b) 63 characters
c) 79 characters
d) none of the mentioned

Answer: d
Explanation: Identifiers can be of any length.

3. Which of the following is invalid?
a) _a = 1
b) __a = 1
c) __str__ = 1
d) none of the mentioned

Answer: d
Explanation: All the statements will execute successfully but at the cost of reduced readability.

4. Which of the following is an invalid variable?
a) my_string_1
b) 1st_string
c) foo

d) _

Answer: b
Explanation: Variable names should not start with a number.

5. Why are local variable names beginning with an underscore discouraged?
a) they are used to indicate a private variables of a class
b) they confuse the interpreter
c) they are used to indicate global variables
d) they slow down execution

Answer: a
Explanation: As Python has no concept of private variables, leading underscores are used to indicate variables that must not be accessed from outside the class.

6. Which of the following is not a keyword?
a) eval
b) assert
c) nonlocal
d) pass

Answer: a
Explanation: eval can be used as a variable.

7. All keywords in Python are in _____
a) lower case
b) UPPER CASE
c) Capitalized
d) None of the mentioned

Answer: d
Explanation: True, False and None are capitalized while the others are in lower case.

8. Which of the following is true for variable names in Python?
a) unlimited length
b) all private members must have leading and trailing underscores
c) underscore and ampersand are the only two special characters allowed
d) none of the mentioned

Answer: a
Explanation: Variable names can be of any length.

9. Which of the following is an invalid statement?
a) abc = 1,000,000
b) a b c = 1000 2000 3000
c) a,b,c = 1000, 2000, 3000

d) a_b_c = 1,000,000

Answer: b
Explanation: Spaces are not allowed in variable names.

10. Which of the following cannot be a variable?
a) __init__
b) in
c) it
d) on

Answer: b
Explanation: in is a keyword.

Python Questions and Answers – Basic Operators

1. Which is the correct operator for power(x^y)?
a) X^y
b) X**y
c) X^^y
d) None of the mentioned

Answer: b
Explanation: In python, power operator is xy i.e. 2**3=8.**

2. Which one of these is floor division?
a) /
b) //
c) %
d) None of the mentioned

Answer: b
Explanation: When both of the operands are integer then python chops out the fraction part and gives you the round off value, to get the accurate answer use floor division. This is floor division. For ex, 5/2 = 2.5 but both of the operands are integer so answer of this expression in python is 2. To get the 2.5 answer, use floor division.

3. What is the order of precedence in python?
i) Parentheses
ii) Exponential
iii) Multiplication
iv) Division
v) Addition
vi) Subtraction
a) i,ii,iii,iv,v,vi
b) ii,i,iii,iv,v,vi
c) ii,i,iv,iii,v,vi
d) i,ii,iii,iv,vi,v

Answer: a
Explanation: For order of precedence, just remember this PEMDAS (similar to BODMAS).

4. What is the answer to this expression, 22 % 3 is?
a) 7
b) 1
c) 0
d) 5

Answer: b
Explanation: Modulus operator gives the remainder. So, 22%3 gives the remainder, that is, 1.

5. Mathematical operations can be performed on a string.
a) True
b) False

Answer: b
Explanation: You can't perform mathematical operation on string even if the string is in the form: '1234…'.

6. Operators with the same precedence are evaluated in which manner?
a) Left to Right
b) Right to Left
c) Can't say
d) None of the mentioned

Answer: a
Explanation: None.

7. What is the output of this expression, 3*1**3?
a) 27
b) 9
c) 3
d) 1

Answer: c
Explanation: First this expression will solve 13 because exponential has higher precedence than multiplication, so 1**3 = 1 and 3*1 = 3. Final answer is 3.**

8. Which one of the following has the same precedence level?
a) Addition and Subtraction
b) Multiplication, Division and Addition
c) Multiplication, Division, Addition and Subtraction
d) Addition and Multiplication

Answer: a
Explanation: "Addition and Subtraction" are at the same precedence level. Similarly, "Multiplication and Division" are at the same precedence level. However, Multiplication and Division operators are at a higher precedence level than Addition and Subtraction operators.

9. The expression Int(x) implies that the variable x is converted to integer.
a) True
b) False

Answer: a
Explanation: None.

10. Which one of the following has the highest precedence in the expression?
a) Exponential
b) Addition
c) Multiplication
d) Parentheses

Answer: d
Explanation: Just remember: PEMDAS, that is, Parenthesis, Exponentiation, Division, Multiplication, Addition, Subtraction. Note that the precedence order of Division and Multiplication is the same. Likewise, the order of Addition and Subtraction is also the same.

Python Questions and Answers – Core Data types

1. Which of these in not a core data type?
a) Lists
b) Dictionary
c) Tuples
d) Class

Answer: d
Explanation: Class is a user defined data type.

2. Given a function that does not return any value, What value is thrown by default when executed in shell.
a) int
b) bool
c) void
d) None

Answer: d
Explanation: Python shell throws a NoneType object back.

3. What will be the output of the following Python code?
1. >>>str="hello"
2. >>>str[:2]
3. >>>
a) he
b) lo
c) olleh
d) hello

Answer: a
Explanation: We are printing only the 1st two bytes of string and hence the answer is "he".

4. Which of the following will run without errors?
a) round(45.8)
b) round(6352.898,2,5)
c) round()
d) round(7463.123,2,1)

Answer: a
Explanation: Execute help(round) in the shell to get details of the parameters that are passed into the round function.

5. What is the return type of function id?
a) int
b) float

c) bool
d) dict

Answer: a
Explanation: Execute help(id) to find out details in python shell.id returns a integer value that is unique.

6. In python we do not specify types, it is directly interpreted by the compiler, so consider the following operation to be performed.
1. >>>x = 13 ? 2
objective is to make sure x has a integer value, select all that apply (python 3.xx)
a) x = 13 // 2
b) x = int(13 / 2)
c) x = 13 % 2
d) All of the mentioned

Answer: d
Explanation: // is integer operation in python 3.0 and int(..) is a type cast operator.

7. What error occurs when you execute the following Python code snippet?
apple = mango
a) SyntaxError
b) NameError
c) ValueError
d) TypeError

Answer: b
Explanation: Mango is not defined hence name error.

8. What will be the output of the following Python code snippet?
```
1.  def example(a):
2.      a = a + '2'
3.       a = a*2
4.      return a
5.  >>>example("hello")
```
a) indentation Error
b) cannot perform mathematical operation on strings
c) hello2
d) hello2hello2

Answer: a
Explanation: Python codes have to be indented properly.

9. What data type is the object below?
L = [1, 23, 'hello', 1]
a) list
b) dictionary

c) array
d) tuple

Answer: a
Explanation: List data type can store any values within it.

10. In order to store values in terms of key and value we use what core data type.
a) list
b) tuple
c) class
d) dictionary

Answer: d
Explanation: Dictionary stores values in terms of keys and values.

11. Which of the following results in a SyntaxError?
a) ‘”Once upon a time…”, she said.’
b) “He said, ‘Yes!’”
c) ‘3\’
d) ”’That’s okay”’

Answer: c
Explanation: Carefully look at the colons.

12. The following is displayed by a print function call. Select all of the function calls that result in this output.
1. tom
2. dick
3. harry

a)
```
print('''tom
\ndick
\nharry''')
```
b) print(”’tomdickharry”’)
c) print(‘tom\ndick\nharry’)
d)
```
print('tom
dick
harry')
```

Answer: c
Explanation: The \n adds a new line.

13. What is the average value of the following Python code snippet?
```
1.  >>>grade1 = 80
2.  >>>grade2 = 90
3.  >>>average = (grade1 + grade2) / 2
```

a) 85.0
b) 85.1
c) 95.0
d) 95.1

Answer: a
Explanation: Cause a decimal value of 0 to appear as output.

14. Select all options that print.
hello-how-are-you
a) print(‘hello’, ‘how’, ‘are’, ‘you’)
b) print(‘hello’, ‘how’, ‘are’, ‘you’ + ‘-‘ * 4)
c) print(‘hello-‘ + ‘how-are-you’)
d) print(‘hello’ + ‘-‘ + ‘how’ + ‘-‘ + ‘are’ + ‘you’)

Answer: c
Explanation: Execute in the shell.

15. What is the return value of trunc()?
a) int
b) bool
c) float
d) None

Answer: a
Explanation: Execute help(math.trunc) to get details.

Python Questions and Answers – Numeric Types

1. What is the output of print 0.1 + 0.2 == 0.3?
a) True
b) False
c) Machine dependent
d) Error

Answer: b
Explanation: Neither of 0.1, 0.2 and 0.3 can be represented accurately in binary. The round off errors from 0.1 and 0.2 accumulate and hence there is a difference of 5.5511e-17 between (0.1 + 0.2) and 0.3.

2. Which of the following is not a complex number?
a) k = 2 + 3j
b) k = complex(2, 3)
c) k = 2 + 3l
d) k = 2 + 3J

Answer: c
Explanation: l (or L) stands for long.

3. What is the type of inf?
a) Boolean
b) Integer
c) Float
d) Complex

Answer: c
Explanation: Infinity is a special case of floating point numbers. It can be obtained by float('inf').

4. What does ~4 evaluate to?
a) -5
b) -4
c) -3
d) +3

Answer: a
Explanation: ~x is equivalent to -(x+1).

5. What does ~~~~~~5 evaluate to?
a) +5
b) -11
c) +11
d) -5

Answer: a
Explanation: ~x is equivalent to -(x+1).

6. Which of the following is incorrect?
a) x = 0b101
b) x = 0x4f5
c) x = 19023
d) x = 03964

Answer: d
Explanation: Numbers starting with a 0 are octal numbers but 9 isn't allowed in octal numbers.

7. What is the result of cmp(3, 1)?
a) 1
b) 0
c) True
d) False

Answer: a
Explanation: cmp(x, y) returns 1 if x > y, 0 if x == y and -1 if x < y.

8. Which of the following is incorrect?
a) float('inf')
b) float('nan')
c) float('56'+'78')
d) float('12+34')

Answer: d
Explanation: '+' cannot be converted to a float.

9. What is the result of round(0.5) – round(-0.5)?
a) 1.0
b) 2.0
c) 0.0
d) Value depends on Python version

Answer: d
Explanation: The behavior of the round() function is different in Python 2 and Python 3. In Python 2, it rounds off numbers away from 0 when the number to be rounded off is exactly halfway through. round(0.5) is 1 and round(-0.5) is -1 whereas in Python 3, it rounds off numbers towards nearest even number when the number to be rounded off is exactly halfway through. See the below output.
Here's the runtime output for Python version 2.7 interpreter.
$ python
Python 2.7.17 (default, Nov 7 2019, 10:07:09)
>>> round(0.5)

1.0
>>> round(-0.5)
-1.0
>>>
In the above output, you can see that the round() functions on 0.5 and -0.5 are moving away from 0 and hence "round(0.5) – (round(-0.5)) = 1 – (-1) = 2"
Here's the runtime output for Python version 3.6 interpreter.
$ python3
Python 3.6.8 (default, Oct 7 2019, 12:59:55)
>>> round(0.5)
0
>>> round(-0.5)
0
>>> round(2.5)
2
>>> round(3.5)
4
>>>
In the above output, you can see that the round() functions on 0.5 and -0.5 are moving towards 0 and hence "round(0.5) – (round(-0.5)) = 0 – 0 = 0". Also note that the round(2.5) is 2 (which is an even number) whereas round(3.5) is 4 (which is an even number).

10. What does 3 ^ 4 evaluate to?
a) 81
b) 12
c) 0.75
d) 7

Answer: d
Explanation: ^ is the Binary XOR operator.

Python Questions and Answers – Operator Precedence and Associativity – 1

1. The value of the expressions 4/(3*(2-1)) and 4/3*(2-1) is the same.
a) True
b) False

Answer: a
Explanation: Although the presence of parenthesis does affect the order of precedence, in the case shown above, it is not making a difference. The result of both of these expressions is 1.333333333. Hence the statement is true.

2. What will be the value of the following Python expression?
4 + 3 % 5
a) 4
b) 7
c) 2
d) 0

Answer: b
Explanation: The order of precedence is: %, +. Hence the expression above, on simplification results in 4 + 3 = 7. Hence the result is 7.

3. Evaluate the expression given below if A = 16 and B = 15.
A % B // A
a) 0.0
b) 0
c) 1.0
d) 1

Answer: b
Explanation: The above expression is evaluated as: 16%15//16, which is equal to 1//16, which results in 0.

4. Which of the following operators has its associativity from right to left?
a) +
b) //
c) %
d) **

Answer: d
Explanation: All of the operators shown above have associativity from left to right, except exponentiation operator (**) which has its associativity from right to left.

5. What will be the value of x in the following Python expression?
x = int(43.55+2/2)
a) 43
b) 44

c) 22
d) 23

Answer: b
Explanation: The expression shown above is an example of explicit conversion. It is evaluated as int(43.55+1) = int(44.55) = 44. Hence the result of this expression is 44.

6. What is the value of the following expression?
2+4.00, 2**4.0
a) (6.0, 16.0)
b) (6.00, 16.00)
c) (6, 16)
d) (6.00, 16.0)

Answer: a
Explanation: The result of the expression shown above is (6.0, 16.0). This is because the result is automatically rounded off to one decimal place.

7. Which of the following is the truncation division operator?
a) /
b) %
c) //
d) |

Answer: c
Explanation: // is the operator for truncation division. It is called so because it returns only the integer part of the quotient, truncating the decimal part. For example: 20//3 = 6.

8. What are the values of the following Python expressions?
2**(3**2)
(2**3)**2
2**3**2
a) 64, 512, 64
b) 64, 64, 64
c) 512, 512, 512
d) 512, 64, 512

Answer: d
Explanation: Expression 1 is evaluated as: 29, which is equal to 512. Expression 2 is evaluated as 8**2, which is equal to 64. The last expression is evaluated as 2**(3**2). This is because the associativity of ** operator is from right to left. Hence the result of the third expression is 512.**

9. What is the value of the following expression?
8/4/2, 8/(4/2)

a) (1.0, 4.0)
b) (1.0, 1.0)
c) (4.0. 1.0)
d) (4.0, 4.0)

Answer: a
Explanation: The above expressions are evaluated as: 2/2, 8/2, which is equal to (1.0, 4.0).

10. What is the value of the following expression?
float(22//3+3/3)
a) 8
b) 8.0
c) 8.3
d) 8.33

Answer: b
Explanation: The expression shown above is evaluated as: float(7+1) = float(8) = 8.0. Hence the result of this expression is 8.0.

Python Questions and Answers – Precedence and Associativity – 2

1. What will be the output of the following Python expression?
print(4.00/(2.0+2.0))
a) Error
b) 1.0
c) 1.00
d) 1

Answer: b
Explanation: The result of the expression shown above is 1.0 because print rounds off digits.

2. What will be the value of X in the following Python expression?
X = 2+9*((3*12)-8)/10
a) 30.0
b) 30.8
c) 28.4
d) 27.2

Answer: d
Explanation: The expression shown above is evaluated as: 2+9*(36-8)/10, which simplifies to give 2+9*(2.8), which is equal to 2+25.2 = 27.2. Hence the result of this expression is 27.2.

3. Which of the following expressions involves coercion when evaluated in Python?
a) 4.7 – 1.5
b) 7.9 * 6.3
c) 1.7 % 2
d) 3.4 + 4.6

Answer: c
Explanation: Coercion is the implicit (automatic) conversion of operands to a common type. Coercion is automatically performed on mixed-type expressions. The expression 1.7 % 2 is evaluated as 1.7 % 2.0 (that is, automatic conversion of int to float).

4. What will be the output of the following Python expression?
24//6%3, 24//4//2
a) (1,3)
b) (0,3)
c) (1,0)
d) (3,1)

Answer: a
Explanation: The expressions are evaluated as: 4%3 and 6//2 respectively. This results in the answer (1,3). This is because the associativity of both of the expressions shown above is left to right.

5. Which among the following list of operators has the highest precedence?
+, -, **, %, /, <<, >>, |
a) <<, >>
b) **
c) |
d) %

Answer: b
Explanation: The highest precedence is that of the exponentiation operator, that is of **.**

6. What will be the value of the following Python expression?
float(4+int(2.39)%2)
a) 5.0
b) 5
c) 4.0
d) 4

Answer: c
Explanation: The above expression is an example of explicit conversion. It is evaluated as: float(4+int(2.39)%2) = float(4+2%2) = float(4+0) = 4.0. Hence the result of this expression is 4.0.

7. Which of the following expressions is an example of type conversion?
a) 4.0 + float(3)
b) 5.3 + 6.3
c) 5.0 + 3
d) 3 + 7

Answer: a
Explanation: Type conversion is nothing but explicit conversion of operands to a specific type. Options 5.3 + 6.3 and 5.0 + 3 are examples of implicit conversion whereas option 4.0 + float(3) is an example of explicit conversion or type conversion.

8. Which of the following expressions results in an error?
a) float(‘10’)
b) int(‘10’)
c) float(’10.8’)
d) int(’10.8’)

Answer: d
Explanation: All of the above examples show explicit conversion. However the expression int(’10.8’) results in an error.

9. What will be the value of the following Python expression?
4+2**5//10

a) 3
b) 7
c) 77
d) 0

Answer: b
Explanation: The order of precedence is: **, //, +. The expression 4+25//10 is evaluated as 4+32//10, which is equal to 4+3 = 7. Hence the result of the expression shown above is 7.**

10. The expression 2**2**3 is evaluates as: (2**2)**3.
a) True
b) False

Answer: b
Explanation: The value of the expression (22)**3 = 4**3 = 64. When the expression 2**2**3 is evaluated in python, we get the result as 256, because this expression is evaluated as 2**(2**3). This is because the associativity of exponentiation operator (**) is from right to left and not from left to right.**

Python Questions and Answers – Bitwise – 1

1. What will be the output of the following Python code snippet if x=1?
x<<2
a) 8
b) 1
c) 2
d) 4

Answer: d
Explanation: The binary form of 1 is 0001. The expression x<<2 implies we are performing bitwise left shift on x. This shift yields the value: 0100, which is the binary form of the number 4.

2. What will be the output of the following Python expression?
bin(29)
a) '0b10111'
b) '0b11101'
c) '0b11111'
d) '0b11011'

Answer: b
Explanation: The binary form of the number 29 is 11101. Hence the output of this expression is '0b11101'.

3. What will be the value of x in the following Python expression, if the result of that expression is 2?
x>>2
a) 8
b) 4
c) 2
d) 1

Answer: a
Explanation: When the value of x is equal to 8 (1000), then x>>2 (bitwise right shift) yields the value 0010, which is equal to 2. Hence the value of x is 8.

4. What will be the output of the following Python expression?
int(1011)?
a) 1011
b) 11
c) 13
d) 1101

Answer: a
Explanation: The result of the expression shown will be 1011. This is because we have not specified the base in this expression. Hence it automatically takes the base as 10.

5. To find the decimal value of 1111, that is 15, we can use the function:
a) int(1111,10)
b) int('1111',10)
c) int(1111,2)
d) int('1111',2)

Answer: d
Explanation: The expression int('1111',2) gives the result 15. The expression int('1111', 10) will give the result 1111.

6. What will be the output of the following Python expression if x=15 and y=12?
x & y
a) b1101
b) 0b1101
c) 12
d) 1101

Answer: c
Explanation: The symbol '&' represents bitwise AND. This gives 1 if both the bits are equal to 1, else it gives 0. The binary form of 15 is 1111 and that of 12 is 1100. Hence on performing the bitwise AND operation, we get 1100, which is equal to 12.

7. Which of the following expressions results in an error?
a) int(1011)
b) int('1011',23)
c) int(1011,2)
d) int('1011')

Answer: c
Explanation: The expression int(1011,2) results in an error. Had we written this expression as int('1011',2), then there would not be an error.

8. Which of the following represents the bitwise XOR operator?
a) &
b) ^
c) |
d) !

Answer: b
Explanation: The ^ operator represent bitwise XOR operation. &: bitwise AND, | : bitwise OR and ! represents bitwise NOT.

9. What is the value of the following Python expression?
bin(0x8)
a) '0bx1000'
b) 8

c) 1000
d) ‘0b1000’

Answer: d
Explanation: The prefix 0x specifies that the value is hexadecimal in nature. When we convert this hexadecimal value to binary form, we get the result as: ‘0b1000’.

10. What will be the output of the following Python expression?
0x35 | 0x75
a) 115
b) 116
c) 117
d) 118

Answer: c
Explanation: The binary value of 0x35 is 110101 and that of 0x75 is 1110101. On OR-ing these two values we get the output as: 1110101, which is equal to 117. Hence the result of the above expression is 117.

Python Questions and Answers – Bitwise – 2

1. It is not possible for the two's complement value to be equal to the original value in any case.
a) True
b) False

Answer: b
Explanation: In most cases the value of two's complement is different from the original value. However, there are cases in which the two's complement value may be equal to the original value. For example, the two's complement of 10000000 is also equal to 10000000. Hence the statement is false.

2. The one's complement of 110010101 is:
a) 001101010
b) 110010101
c) 001101011
d) 110010100

Answer: a
Explanation: The one's complement of a value is obtained by simply changing all the 1's to 0's and all the 0's to 1's. Hence the one's complement of 110010101 is 001101010.

3. Bitwise _____ gives 1 if either of the bits is 1 and 0 when both of the bits are 1.
a) OR
b) AND
c) XOR
d) NOT

Answer: c
Explanation: Bitwise XOR gives 1 if either of the bits is 1 and 0 when both of the bits are 1.

4. What will be the output of the following Python expression?
`4^12`
a) 2
b) 4
c) 8
d) 12

Answer: c
Explanation: ^ is the XOR operator. The binary form of 4 is 0100 and that of 12 is 1100. Therefore, 0100^1100 is 1000, which is equal to 8.

5. Any odd number on being AND-ed with _____ always gives 1. Hint: Any even number on being AND-ed with this value always gives 0.

a) 10
b) 2
c) 1
d) 0

Answer: c
Explanation: Any odd number on being AND-ed with 1 always gives 1. Any even number on being AND-ed with this value always gives 0.

6. What will be the value of the following Python expression?
bin(10-2)+bin(12^4)
a) 0b10000
b) 0b10001000
c) 0b1000b1000
d) 0b10000b1000

Answer: d
Explanation: The output of bin(10-2) = 0b1000 and that of bin(12^4) is ob1000. Hence the output of the above expression is: 0b10000b1000.

7. Which of the following expressions can be used to multiply a given number 'a' by 4?
a) a<<2
b) a<<4
c) a>>2
d) a>>4

Answer: a
Explanation: Let us consider an example wherein a=2. The binary form of 2 is 0010. When we left shift this value by 2, we get 1000, the value of which is 8. Hence if we want to multiply a given number 'a' by 4, we can use the expression: a<<2.

8. What will be the output of the following Python code if a=10 and b =20?
```
a=10
b=20
a=a^b
b=a^b
a=a^b
print(a,b)
```
a) 10 20
b) 10 10
c) 20 10
d) 20 20

Answer: c
Explanation: The code shown above is used to swap the contents of two memory locations using bitwise X0R operator. Hence the output of the code shown above is: 20 10.

9. What is the two's complement of -44?
a) 1011011
b) 11010100
c) 11101011
d) 10110011

Answer: b
Explanation: The binary form of -44 is 00101100. The one's complement of this value is 11010011. On adding one to this we get: 11010100 (two's complement).

10. What will be the output of the following Python expression?
~100?
a) 101
b) -101
c) 100
d) -100

Answer: b
Explanation: Suppose we have an expression ~A. This is evaluated as: -A – 1. Therefore, the expression ~100 is evaluated as -100 – 1, which is equal to -101.

Python Questions and Answers – Boolean

1. What will be the output of the following Python code snippet?

```
bool('False')
bool()
```

a)
True
True

b)
False
True

c)
False
False

d)
True
False

Answer: d

Explanation: The Boolean function returns true if the argument passed to the bool function does not amount to zero. In the first example, the string 'False' is passed to the function bool. This does not amount to zero and hence the output is true. In the second function, an empty list is passed to the function bool. Hence the output is false.

2. What will be the output of the following Python code snippet?

```
['hello', 'morning'][bool('')]
```

a) error
b) no output
c) hello
d) morning

Answer: c

Explanation: The line of code shown above can be simplified to state that 'hello' should be printed if the argument passed to the Boolean function amounts to zero, else 'morning' will be printed.

3. What will be the output of the following Python code snippet?

```
not(3>4)
not(1&1)
```

a)
True
True

b)
True
False

c)
False

True
d)
False
False

Answer: b
Explanation: The function not returns true if the argument amounts to false, and false if the argument amounts to true. Hence the first function returns false, and the second function returns false.

4. What will be the output of the following Python code?
['f', 't'][bool('spam')]
a) t
b) f
c) No output
d) Error

Answer: a
Explanation: The line of code can be translated to state that 'f' is printed if the argument passed to the Boolean function amount to zero. Else 't' is printed. The argument given to the Boolean function in the above case is 'spam', which does not amount to zero. Hence the output is t.

5. What will be the output of the following Python code?
l=[1, 0, 2, 0, 'hello', '', []]
list(filter(bool, l))
a) Error
b) [1, 0, 2, 0, 'hello', '', []]
c) [1, 0, 2, 'hello', '', []]
d) [1, 2, 'hello']

Answer: d
Explanation: The code shown above returns a new list containing only those elements of the list l which do not amount to zero. Hence the output is: [1, 2, 'hello'].

6. What will be the output of the following Python code if the system date is 21st June, 2017 (Wednesday)?
[] or {}
{} or []
a)
[]
{}
b)
[]
[]
c)
{}

[]
d)
{}
{}

Answer: c
Explanation: The code shown above shows two functions. In both the cases the right operand is returned. This is because each function is evaluated from left to right. Since the left operand is false, it is assumed that the right operand must be true and hence the right operand is returned in each of the above case.

7. What will be the output of the following Python code?

```
class Truth:
        pass
x=Truth()
bool(x)
```

a) pass
b) true
c) false
d) error

Answer: b
Explanation: If the truth method is not defined, the object is considered true. Hence the output of the code shown above is true.

8. What will be the output of the following Python code?

```
if (9 < 0) and (0 < -9):
    print("hello")
elif (9 > 0) or False:
    print("good")
else:
    print("bad")
```

a) error
b) hello
c) good
d) bad

Answer: c
Explanation: The code shown above prints the appropriate option depending on the conditions given. The condition which matches is (9>0), and hence the output is: good.

9. Which of the following Boolean expressions is not logically equivalent to the other three?
a) not(-6<0 or-6>10)
b) -6>=0 and -6<=10
c) not(-6<10 or-6==10)

d) not(-6>10 or-6==10)

Answer: d
Explanation: The expression not(-6<0 or -6>10) returns the output False.
The expression -6>=0 and -6<=10 returns the output False.
The expression not(-6<10 or -6==10) returns the output False.
The expression not(-6>10 or -6==10) returns the output True.

10. What will be the output of the following Python code snippet?
not(10<20) and not(10>30)
a) True
b) False
c) Error
d) No output

Answer: b
Explanation: The expression not(10<20) returns false. The expression not(10>30) returns true. The and operation between false and true returns false. Hence the output is false.

Python Question and Answers – Formatting – 1

1. What will be the output of the following Python code snippet?
X=”hi”
print(“05d”%X)
a) 00000hi
b) 000hi
c) hi000
d) error

Answer: d
Explanation: The code snippet shown above results in an error because the above formatting option works only if ‘X’ is a number. Since in the above case ‘X’ is a string, an error is thrown.

2. What will be the output of the following Python code snippet?
X=”san-foundry”
print(“%56s”,X)
a) 56 blank spaces before san-foundry
b) 56 blank spaces before san and foundry
c) 56 blank spaces after san-foundry
d) no change

Answer: a
Explanation: The formatting option print(“%Ns”,X) helps us add ‘N’ number of spaces before a given string ‘X’. Hence the output for the code snippet shown above will be 56 blank spaces before the string “san-foundry”.

3. What will be the output of the following Python expression if x=456?
print("%-06d"%x)
a) 000456
b) 456000
c) 456
d) error

Answer: c
Explanation: The expression shown above results in the output 456.

4. What will be the output of the following Python expression if X=345?
print(“%06d”%X)
a) 345000
b) 000345
c) 000000345
d) 345000000

Answer: b
Explanation: The above expression returns the output 000345. It adds the required

number of zeroes before the given number in order to make the number of digits 6 (as specified in this case).

5. Which of the following formatting options can be used in order to add ‘n’ blank spaces after a given string ‘S’?
a) print(“-ns”%S)
b) print(“-ns”%S)
c) print(“%ns”%S)
d) print(“%-ns”%S)

Answer: d
Explanation: In order to add ‘n’ blank spaces after a given string ‘S’, we use the formatting option:(“%-ns”%S).

6. What will be the output of the following Python expression if X = -122?
print("-%06d"%x)
a) -000122
b) 000122
c) –00122
d) -00122

Answer: c
Explanation: The given number is -122. Here the total number of digits (including the negative sign) should be 6 according to the expression. In addition to this, there is a negative sign in the given expression. Hence the output will be – -00122.

7. What will be the output of the following Python expression if the value of x is 34?
print(“%f”%x)
a) 34.00
b) 34.0000
c) 34.000000
d) 34.00000000

Answer: c
Explanation: The expression shown above normally returns the value with 6 decimal points if it is not specified with any number. Hence the output of this expression will be: 34.000000 (6 decimal points).

8. What will be the output of the following Python expression if x=56.236?
print("%.2f"%x)
a) 56.00
b) 56.24
c) 56.23
d) 0056.236

Answer: b
Explanation: The expression shown above rounds off the given number to the number

of decimal places specified. Since the expression given specifies rounding off to two decimal places, the output of this expression will be 56.24. Had the value been x=56.234 (last digit being any number less than 5), the output would have been 56.23.

9. What will be the output of the following Python expression if x=22.19?
print("%5.2f"%x)
a) 22.1900
b) 22.00000
c) 22.19
d) 22.20

Answer: c
Explanation: The output of the expression above will be 22.19. This expression specifies that the total number of digits (including the decimal point) should be 5, rounded off to two decimal places.

10. The expression shown below results in an error.
print("-%5d0",989)
a) True
b) False

Answer: b
Explanation: The expression shown above does not result in an error. The output of this expression is -%5d0 989. Hence this statement is incorrect.

Python Question and Answers – Formatting – 2

1. What will be the output of the following Python code snippet?
'%d %s %g you' %(1, 'hello', 4.0)
a) Error
b) 1 hello you 4.0
c) 1 hello 4 you
d) 1 4 hello you

Answer: c
Explanation: In the snippet of code shown above, three values are inserted into the target string. When we insert more than one value, we should group the values on the right in a tuple. The % formatting expression operator expects either a single item or a tuple of one or more items on its right side.

2. The output of which of the codes shown below will be: "There are 4 blue birds."?
a) 'There are %g %d birds.' %4 %blue
b) 'There are %d %s birds.' %(4, blue)
c) 'There are %s %d birds.' %[4, blue]
d) 'There are %d %s birds.' 4, blue

Answer: b
Explanation: The code 'There are %d %s birds.' %(4, blue) results in the output: There are 4 blue birds. When we insert more than one value, we should group the values on the right in a tuple.

3. What will be the output of the python code shown below for various styles of format specifiers?
x=1234
res='integers:...%d...%-6d...%06d' %(x, x, x)
res
a) 'integers:…1234…1234 …001234'
b) 'integers…1234…1234…123400'
c) 'integers:… 1234…1234…001234'
d) 'integers:…1234…1234…001234'

Answer: a
Explanation: The code shown above prints 1234 for the format specified %d, '1234 ' for the format specifier %-6d (minus '-' sign signifies left justification), and 001234 for the format specifier %06d. Hence the output of this code is: 'integers:…1234…1234 …001234'

4. What will be the output of the following Python code snippet?
x=3.3456789
'%f | %e | %g' %(x, x, x)
a) Error
b) '3.3456789 | 3.3456789+00 | 3.345678'

c) ‘3.345678 | 3.345678e+0 | 3.345678’
d) ‘3.345679 | 3.345679e+00 | 3.34568’

Answer: d
Explanation: The %f %e and %g format specifiers represent floating point numbers in different ways. %e and %E are the same, except that the exponent is in lowercase. %g chooses the format by number content. Hence the output of this code is: ‘3.345679 | 3.345679e+00 | 3.34568’.

5. What will be the output of the following Python code snippet?
```
x=3.3456789
'%-6.2f | %05.2f | %+06.1f' %(x, x, x)
```
a) ‘3.35 | 03.35 | +003.3’
b) ‘3.3456789 | 03.3456789 | +03.3456789’
c) Error
d) ‘3.34 | 03.34 | 03.34+’

Answer: a
Explanation: The code shown above rounds the floating point value to two decimal places. In this code, a variety of addition formatting features such as zero padding, total field width etc. Hence the output of this code is: ‘3.35 | 03.35 | +003.3’.

6. What will be the output of the following Python code snippet?
```
x=3.3456789
'%s' %x, str(x)
```
a) Error
b) (‘3.3456789’, ‘3.3456789’)
c) (3.3456789, 3.3456789)
d) (‘3.3456789’, 3.3456789)

Answer: b
Explanation: We can simply convert strings with a %s format expression or the str built-in function. Both of these methods have been shown in this code. Hence the output is:) (‘3.3456789’, ‘3.3456789’)

7. What will be the output of the following Python code snippet?
```
'%(qty)d more %(food)s' %{'qty':1, 'food': 'spam'}
```
a) Error
b) No output
c) ‘1 more foods’
d) ‘1 more spam’

Answer: d
Explanation: String formatting also allows conversion targets on the left to refer to the keys in a dictionary coded on the right and fetch the corresponding values. In the code shown above, (qty) and (food) in the format string on the left refers to keys in the

dictionary literal on the right and fetch their assorted values. Hence the output of the code shown above is: 1 more spam.

8. What will be the output of the following Python code snippet?
```
a='hello'
q=10
vars()
```
a) {'a' : 'hello', 'q' : 10, ……..plus built-in names set by Python….}
b) {……Built in names set by Python……}
c) {'a' : 'hello', 'q' : 10}
d) Error

Answer: a
Explanation: The built in function vars() returns a dictionary containing all the variables that exist in the place. Hence the output of the code shown above is: {'a' : 'hello', 'q' : 10, ……..plus built-in names set by Python….}

9. What will be the output of the following Python code?
```
s='{0}, {1}, and {2}'
s.format('hello', 'good', 'morning')
```
a) 'hello good and morning'
b) 'hello, good, morning'
c) 'hello, good, and morning'
d) Error

Answer: c
Explanation: Within the subject string, curly braces designate substitution targets and arguments to be inserted either by position or keyword. Hence the output of the code shown above:'hello, good,and morning'.

10. What will be the output of the following Python code?
```
s='%s, %s & %s'
s%('mumbai', 'kolkata', 'delhi')
```
a) mumbai kolkata & delhi
b) Error
c) No output
d) 'mumbai, kolkata & delhi'

Answer: d
Explanation: In the code shown above, the format specifier %s is replaced by the designated substitution. Hence the output of the code shown above is: 'mumbai, kolkata & delhi'.

11. What will be the output of the following Python code?
```
t = '%(a)s, %(b)s, %(c)s'
t % dict(a='hello', b='world', c='universe')
```

a) ‘hello, world, universe’
b) ‘hellos, worlds, universes’
c) Error
d) hellos, world, universe

Answer: a
Explanation: Within the subject string, curly braces represent substitution targets and arguments to be inserted. Hence the output of the code shown above: ‘hello, world, universe’.

12. What will be the output of the following Python code?
'{a}, {0}, {abc}'.format(10, a=2.5, abc=[1, 2])
a) Error
b) ‘2.5, 10, [1, 2]’
c) 2.5, 10, 1, 2
d) ’10, 2.5, [1, 2]’

Answer: b
Explanation: Since we have specified that the order of the output be: {a}, {0}, {abc}, hence the value of associated with {a} is printed first followed by that of {0} and {abc}. Hence the output of the code shown above is: ‘2.5, 10, [1, 2]’.

13. What will be the output of the following Python code?
'{0:.2f}'.format(1.234)
a) ‘1’
b) ‘1.234’
c) ‘1.23’
d) ‘1.2’

Answer: c
Explanation: The code shown above displays the string method to round off a given decimal number to two decimal places. Hence the output of the code is: ‘1.23’.

14. What will be the output of the following Python code?
'%x %d' %(255, 255)
a) ‘ff, 255’
b) ‘255, 255’
c) ‘15f, 15f’
d) Error

Answer: a
Explanation: The code shown above converts the given arguments to hexadecimal and decimal values and prints the result. This is done using the format specifiers %x and %d respectively. Hence the output of the code shown above is: ‘ff, 255’.

15. The output of the two codes shown below is the same.
i. '{0:.2f}'.format(1/3.0)

ii. '%.2f'%(1/3.0)
a) True
b) False

Answer: a
Explanation: The two codes shown above represent the same operation but in different formats. The output of both of these functions is: '0.33'. Hence the statement is true.

Python Questions and Answers – Advanced Formatting Tools

1. What will be the output of the following Python code?
```
l=list('HELLO')
'first={0[0]}, third={0[2]}'.format(l)
```
a) ‘first=H, third=L’
b) ‘first=0, third=2’
c) Error
d) ‘first=0, third=L’

Answer: a
Explanation: In the code shown above, the value for first is substituted by l[0], that is H and the value for third is substituted by l[2], that is L. Hence the output of the code shown above is: ‘first=H, third=L’. The list l= [‘H’, ‘E’, ‘L’, ‘L’, ‘O’].

2. What will be the output of the following Python code?
```
l=list('HELLO')
p=l[0], l[-1], l[1:3]
'a={0}, b={1}, c={2}'.format(*p)
```
a) Error
b) “a=’H’, b=’O’, c=(E, L)”
c) “a=H, b=O, c=[‘E’, ‘L’]”
d) Junk value

Answer: c
Explanation: In the code shown above, the value for a is substituted by l[0], that is ‘H’, the value of b is substituted by l[-1], that is ‘O’ and the value for c is substituted by l[1:3]. Here the use of *p is to unpack a tuple items into individual function arguments.

3. The formatting method {1:<10} represents the _____ positional argument, _____ justified in a 10 character wide field.
a) first, right
b) second, left
c) first, left
d) second, right

Answer: b
Explanation: The formatting method {1:<10} represents the second positional argument, left justified in a 10 character wide field.

4. What will be the output of the following Python code?
```
hex(255), int('FF', 16), 0xFF
```
a) [0xFF, 255, 16, 255]
b) (‘0xff’, 155, 16, 255)
c) Error

d) (‘0xff’, 255, 255)

Answer: d
Explanation: The code shown above converts the value 255 into hexadecimal, that is, 0xff. The value ‘FF’ into integer. Hence the output of the code shown is: (‘0xff’, 255, 255).

5. The output of the two codes shown below is the same.
i. bin((2**16)-1)
ii. '{}'.format(bin((2**16)-1))
a) True
b) False

Answer: a
Explanation: The output of both of the codes shown above is ‘0b1111111111111111’. Hence the statement is true.

6. What will be the output of the following Python code?
'{a}{b}{a}'.format(a='hello', b='world')
a) ‘hello world’
b) ‘hello’ ‘world’ ‘hello’
c) ‘helloworldhello’
d) ‘hello’ ‘hello’ ‘world’

Answer: c
Explanation: The code shown above prints the values substituted for a, b, a, in the same order. This operation is performed using the format function. Hence the output of the code is: ‘helloworldhello’.

7. What will be the output of the following Python code?
D=dict(p='san', q='foundry')
'{p}{q}'.format(**D)
a) Error
b) sanfoundry
c) san foundry
d) {‘san’, ‘foundry’}

Answer: b
Explanation: The code shown above prints the values substituted for p and q in the same order. Note that there is no blank space between p and q. Hence the output is: sanfoundry.

8. What will be the output of the following Python code?
'The {} side {1} {2}'.format('bright', 'of', 'life')
a) Error
b) ‘The bright side of life’
c) ‘The {bright} side {of} {life}’

d) No output

Answer: a
Explanation: The code shown above results in an error. This is because we have switched from automatic field numbering to manual field numbering, that is, from {} to {1}. Hence this code results in an error.

9. What will be the output of the following Python code?
'{0:f}, {1:2f}, {2:05.2f}'.format(1.23456, 1.23456, 1.23456)
a) Error
b) '1.234560, 1.22345, 1.23'
c) No output
d) '1.234560, 1.234560, 01.23'

Answer: d
Explanation: In the code shown above, various formatting options are displayed using the format option. Hence the output of this code is: '1.234560, 1.234560, 01.23'

10. What will be the output of the following Python code?
'%.2f%s' % (1.2345, 99)
a) '1.2345', '99'
b) '1.2399'
c) '1.234599'
d) 1.23, 99

Answer: b
Explanation: In this code, we must notice that since multiple values haven been given, they should be enclosed in a tuple. Since the formatting format is %.2f, the value 1.2345 is reduced to two decimal places. Hence the output of the code shown above: '1.2399'.

11. What will be the output of the following Python code?
'%s' %((1.23,),)
a) '(1.23,)'
b) 1.23,
c) (,1.23)
d) '1.23'

Answer: a
Explanation: The formatting expression accepts either a single substitution value, or a tuple of one or more items. Since single item can be given either by itself or within the tuple, a tuple to be formatted must be provided as a tested tuple. Hence the output of the code is: >>> '%s' %((1.23,),).

12. What will be the output of the following two codes?
i. '{0}'.format(4.56)
ii. '{0}'.format([4.56,])

a) ‘4.56’, ‘4.56,’
b) ‘4.56’, ‘[4.56]’
c) 4.56, [4.56,]
d) 4.56, [4.56,]

Answer: b
Explanation: The code shown above shows the formatting option on the same value, that is 4.56, where in the second case, the value is enclosed in a list. Hence the output of the code shown above is:
‘4.56’, ‘[4.56]’

Python Questions and Answers – While and For Loops – 1

1. What will be the output of the following Python code?

```
x = ['ab', 'cd']
for i in x:
    i.upper()
print(x)
```

a) [‘ab’, ‘cd’]
b) [‘AB’, ‘CD’]
c) [None, None]
d) none of the mentioned

Answer: a
Explanation: The function upper() does not modify a string in place, it returns a new string which isn’t being stored anywhere.

2. What will be the output of the following Python code?

```
x = ['ab', 'cd']
for i in x:
    x.append(i.upper())
print(x)
```

a) [‘AB’, ‘CD’]
b) [‘ab’, ‘cd’, ‘AB’, ‘CD’]
c) [‘ab’, ‘cd’]
d) none of the mentioned

Answer: d
Explanation: The loop does not terminate as new elements are being added to the list in each iteration.

3. What will be the output of the following Python code?

```
i = 1
while True:
    if i%3 == 0:
        break
    print(i)

    i + = 1
```

a) 1 2
b) 1 2 3
c) error
d) none of the mentioned

Answer: c
Explanation: SyntaxError, there shouldn’t be a space between + and = in +=.

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

```
i = 1
while True:
   if i%0O7 == 0:
      break
   print(i)
   i += 1
```
a) 1 2 3 4 5 6
b) 1 2 3 4 5 6 7
c) error
d) none of the mentioned

Answer: a
Explanation: Control exits the loop when i becomes 7.

5. What will be the output of the following Python code?
```
i = 5
while True:
   if i%0O11 == 0:
      break
   print(i)
   i += 1
```
a) 5 6 7 8 9 10
b) 5 6 7 8
c) 5 6
d) error

Answer: b
Explanation: 0O11 is an octal number.

6. What will be the output of the following Python code?
```
i = 5
while True:
   if i%0O9 == 0:
      break
   print(i)
   i += 1
```
a) 5 6 7 8
b) 5 6 7 8 9
c) 5 6 7 8 9 10 11 12 13 14 15 ….
d) error

Answer: d
Explanation: 9 isn’t allowed in an octal number.

7. What will be the output of the following Python code?
```
i = 1
while True:
```

```
    if i%2 == 0:
        break
    print(i)
    i += 2
```
a) 1
b) 1 2
c) 1 2 3 4 5 6 …
d) 1 3 5 7 9 11 …

Answer: d
Explanation: The loop does not terminate since i is never an even number.

8. What will be the output of the following Python code?
```
i = 2
while True:
    if i%3 == 0:
        break
    print(i)
    i += 2
```
a) 2 4 6 8 10 …
b) 2 4
c) 2 3
d) error

Answer: b
Explanation: The numbers 2 and 4 are printed. The next value of i is 6 which is divisible by 3 and hence control exits the loop.

9. What will be the output of the following Python code?
```
i = 1
while False:
    if i%2 == 0:
        break
    print(i)
    i += 2
```
a) 1
b) 1 3 5 7 …
c) 1 2 3 4 …
d) none of the mentioned

Answer: d
Explanation: Control does not enter the loop because of False.

10. What will be the output of the following Python code?
```
True = False
while True:
    print(True)
```

break
a) True
b) False
c) None
d) none of the mentioned

Answer: d
Explanation: SyntaxError, True is a keyword and it's value cannot be changed.

Python Questions and Answers – While and For Loops – 2

1. What will be the output of the following Python code?

```
i = 0
while i < 5:
    print(i)
    i += 1
    if i == 3:
        break
else:
    print(0)
```

a) 0 1 2 0
b) 0 1 2
c) error
d) none of the mentioned

Answer: b
Explanation: The else part is not executed if control breaks out of the loop.

2. What will be the output of the following Python code?

```
i = 0
while i < 3:
    print(i)
    i += 1
else:
    print(0)
```

a) 0 1 2 3 0
b) 0 1 2 0
c) 0 1 2
d) error

Answer: b
Explanation: The else part is executed when the condition in the while statement is false.

3. What will be the output of the following Python code?

```
x = "abcdef"
while i in x:
    print(i, end=" ")
```

a) a b c d e f
b) abcdef
c) i i i i i i …
d) error

Answer: d
Explanation: NameError, i is not defined.

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

```
x = "abcdef"
i = "i"
while i in x:
    print(i, end=" ")
```

a) no output
b) i i i i i i …
c) a b c d e f
d) abcdef

Answer: a
Explanation: “i” is not in “abcdef”.

5. What will be the output of the following Python code?

```
x = "abcdef"
i = "a"
while i in x:
    print(i, end = " ")
```

a) no output
b) i i i i i i …
c) a a a a a a …
d) a b c d e f

Answer: c
Explanation: As the value of i or x isn’t changing, the condition will always evaluate to True.

6. What will be the output of the following Python code?

```
x = "abcdef"
i = "a"
while i in x:
    print('i', end = " ")
```

a) no output
b) i i i i i i …
c) a a a a a a …
d) a b c d e f

Answer: b
Explanation: As the value of i or x isn’t changing, the condition will always evaluate to True.

7. What will be the output of the following Python code?

```
x = "abcdef"
i = "a"
while i in x:
    x = x[:-1]
    print(i, end = " ")
```

a) i i i i i i
b) a a a a a a
c) a a a a a
d) none of the mentioned

Answer: b
Explanation: The string x is being shortened by one character in each iteration.

8. What will be the output of the following Python code?
```
x = "abcdef"
i = "a"
while i in x[:-1]:
    print(i, end = " ")
```
a) a a a a a
b) a a a a a a
c) a a a a a a …
d) a

Answer: c
Explanation: String x is not being altered and i is in x[:-1].

9. What will be the output of the following Python code?
```
x = "abcdef"
i = "a"
while i in x:
    x = x[1:]
    print(i, end = " ")
```
a) a a a a a a
b) a
c) no output
d) error

Answer: b
Explanation: The string x is being shortened by one character in each iteration.

10. What will be the output of the following Python code?
```
x = "abcdef"
i = "a"
while i in x[1:]:
    print(i, end = " ")
```
a) a a a a a a
b) a
c) no output
d) error

Answer: c
Explanation: i is not in x[1:].

Python Questions and Answers – While and For Loops – 3

1. What will be the output of the following Python code?

```
x = 'abcd'
for i in x:
    print(i)
    x.upper()
```

a) a B C D
b) a b c d
c) A B C D
d) error

Answer: b
Explanation: Changes do not happen in-place, rather a new instance of the string is returned.

2. What will be the output of the following Python code?

```
x = 'abcd'
for i in x:
    print(i.upper())
```

a) a b c d
b) A B C D
c) a B C D
d) error

Answer: b
Explanation: The instance of the string returned by upper() is being printed.

3. What will be the output of the following Python code?

```
x = 'abcd'
for i in range(x):
    print(i)
```

a) a b c d
b) 0 1 2 3
c) error
d) none of the mentioned

Answer: c
Explanation: range(str) is not allowed.

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

```
x = 'abcd'
for i in range(len(x)):
    print(i)
```

a) a b c d
b) 0 1 2 3
c) error

d) 1 2 3 4

Answer: b
Explanation: i takes values 0, 1, 2 and 3.

5. What will be the output of the following Python code?
```
x = 'abcd'
for i in range(len(x)):
    print(i.upper())
```
a) a b c d
b) 0 1 2 3
c) error
d) 1 2 3 4

Answer: c
Explanation: Objects of type int have no attribute upper().

6. What will be the output of the following Python code snippet?
```
x = 'abcd'
for i in range(len(x)):
    i.upper()
print (x)
```
a) a b c d
b) 0 1 2 3
c) error
d) none of the mentioned

Answer: c
Explanation: Objects of type int have no attribute upper().

7. What will be the output of the following Python code snippet?
```
x = 'abcd'
for i in range(len(x)):
    x[i].upper()
print (x)
```
a) abcd
b) ABCD
c) error
d) none of the mentioned

Answer: a
Explanation: Changes do not happen in-place, rather a new instance of the string is returned.

8. What will be the output of the following Python code snippet?
```
x = 'abcd'
for i in range(len(x)):
```

```
    i[x].upper()
print (x)
```
a) abcd
b) ABCD
c) error
d) none of the mentioned

Answer: c
Explanation: Objects of type int aren't subscriptable. However, if the statement was x[i], an error would not have been thrown.

9. What will be the output of the following Python code snippet?
```
x = 'abcd'
for i in range(len(x)):
    x = 'a'
    print(x)
```
a) a
b) abcd abcd abcd
c) a a a a
d) none of the mentioned

Answer: c
Explanation: range() is computed only at the time of entering the loop.

10. What will be the output of the following Python code snippet?
```
x = 'abcd'
for i in range(len(x)):
    print(x)
    x = 'a'
```
a) a
b) abcd abcd abcd abcd
c) a a a a
d) none of the mentioned

Answer: d
Explanation: abcd a a a is the output as x is modified only after 'abcd' has been printed once.

Python Questions and Answers – While and For Loops – 4

1. What will be the output of the following Python code?
```
x = 123
for i in x:
    print(i)
```
a) 1 2 3
b) 123
c) error
d) none of the mentioned

Answer: c
Explanation: Objects of type int are not iterable.

2. What will be the output of the following Python code?
```
d = {0: 'a', 1: 'b', 2: 'c'}
for i in d:
    print(i)
```
a) 0 1 2
b) a b c
2 c 1 b c) 0 a
d) none of the mentioned

Answer: a
Explanation: Loops over the keys of the dictionary.

3. What will be the output of the following Python code?
```
d = {0: 'a', 1: 'b', 2: 'c'}
for x, y in d:
    print(x, y)
```
a) 0 1 2
b) a b c
2 c 1 b c) 0 a
d) none of the mentioned

Answer: d
Explanation: Error, objects of type int aren't iterable.

4. What will be the output of the following Python code?
```
d = {0: 'a', 1: 'b', 2: 'c'}
for x, y in d.items():
    print(x, y)
```
a) 0 1 2
b) a b c
2 c 1 b c) 0 a
d) none of the mentioned

Answer: c
Explanation: Loops over key, value pairs.

5. What will be the output of the following Python code?
```
d = {0: 'a', 1: 'b', 2: 'c'}
for x in d.keys():
    print(d[x])
```
a) 0 1 2
b) a b c
2 c 1 b c) 0 a
d) none of the mentioned

Answer: b
Explanation: Loops over the keys and prints the values.

6. What will be the output of the following Python code?
```
d = {0: 'a', 1: 'b', 2: 'c'}
for x in d.values():
    print(x)
```
a) 0 1 2
b) a b c
2 c 1 b c) 0 a
d) none of the mentioned

Answer: b
Explanation: Loops over the values.

7. What will be the output of the following Python code?
```
d = {0: 'a', 1: 'b', 2: 'c'}
for x in d.values():
    print(d[x])
```
a) 0 1 2
b) a b c
2 c 1 b c) 0 a
d) none of the mentioned

Answer: d
Explanation: Causes a KeyError.

8. What will be the output of the following Python code?
```
d = {0, 1, 2}
for x in d.values():
    print(x)
```
a) 0 1 2
b) None None None
c) error

d) none of the mentioned

Answer: c
Explanation: Objects of type set have no attribute values.

9. What will be the output of the following Python code?
```
d = {0, 1, 2}
for x in d:
    print(x)
```
a) 0 1 2
b) {0, 1, 2} {0, 1, 2} {0, 1, 2}
c) error
d) none of the mentioned

Answer: a
Explanation: Loops over the elements of the set and prints them.

10. What will be the output of the following Python code?
```
d = {0, 1, 2}
for x in d:
    print(d.add(x))
```
a) 0 1 2
b) 0 1 2 0 1 2 0 1 2 …
c) None None None
d) None of the mentioned

Answer: c
Explanation: Variable x takes the values 0, 1 and 2. set.add() returns None which is printed.

11. What will be the output of the following Python code?
```
for i in range(0):
    print(i)
```
a) 0
b) no output
c) error
d) none of the mentioned

Answer: b
Explanation: range(0) is empty.

Python Questions and Answers – While and For Loops – 5

1. What will be the output of the following Python code?
```
for i in range(2.0):
    print(i)
```
a) 0.0 1.0
b) 0 1
c) error
d) none of the mentioned

Answer: c
Explanation: Object of type float cannot be interpreted as an integer.

2. What will be the output of the following Python code?
```
for i in range(int(2.0)):
    print(i)
```
a) 0.0 1.0
b) 0 1
c) error
d) none of the mentioned

Answer: b
Explanation: range(int(2.0)) is the same as range(2).

3. What will be the output of the following Python code?
```
for i in range(float('inf')):
    print (i)
```
a) 0.0 0.1 0.2 0.3 …
b) 0 1 2 3 …
c) 0.0 1.0 2.0 3.0 …
d) none of the mentioned

Answer: d
Explanation: Error, objects of type float cannot be interpreted as an integer.

4. What will be the output of the following Python code?
```
for i in range(int(float('inf'))):
    print (i)
```
a) 0.0 0.1 0.2 0.3 …
b) 0 1 2 3 …
c) 0.0 1.0 2.0 3.0 …
d) none of the mentioned

Answer: d
Explanation: OverflowError, cannot convert float infinity to integer.

5. What will be the output of the following Python code snippet?

```
for i in [1, 2, 3, 4][::-1]:
    print (i)
```

a) 1 2 3 4
b) 4 3 2 1
c) error
d) none of the mentioned

Answer: b
Explanation: [::-1] reverses the list.

6. What will be the output of the following Python code snippet?

```
for i in ''.join(reversed(list('abcd'))):
    print (i)
```

a) a b c d
b) d c b a
c) error
d) none of the mentioned

Answer: b
Explanation: ' '.join(reversed(list('abcd'))) reverses a string.

7. What will be the output of the following Python code snippet?

```
for i in 'abcd'[::-1]:
    print (i)
```

a) a b c d
b) d c b a
c) error
d) none of the mentioned

Answer: b
Explanation: [::-1] reverses the string.

8. What will be the output of the following Python code snippet?

```
for i in '':
    print (i)
```

a) None
b) (nothing is printed)
c) error
d) none of the mentioned

Answer: b
Explanation: The string does not have any character to loop over.

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

```
x = 2
for i in range(x):
```

```
    x += 1
    print (x)
```
a) 0 1 2 3 4 …
b) 0 1
c) 3 4
d) 0 1 2 3

Answer: c
Explanation: Variable x is incremented and printed twice.

10. What will be the output of the following Python code snippet?
```
x = 2
for i in range(x):
    x -= 2
    print (x)
```
a) 0 1 2 3 4 …
b) 0 -2
c) 0
d) error

Answer: b
Explanation: The loop is entered twice.

Python Questions and Answers – While and For Loops – 6

1. What will be the output of the following Python code?

```
for i in range(10):
    if i == 5:
        break
    else:
        print(i)
else:
    print("Here")
```

a) 0 1 2 3 4 Here
b) 0 1 2 3 4 5 Here
c) 0 1 2 3 4
d) 1 2 3 4 5

Answer: c
Explanation: The else part is executed if control doesn't break out of the loop.

2. What will be the output of the following Python code?

```
for i in range(5):
    if i == 5:
        break
    else:
        print(i)
else:
    print("Here")
```

a) 0 1 2 3 4 Here
b) 0 1 2 3 4 5 Here
c) 0 1 2 3 4
d) 1 2 3 4 5

Answer: a
Explanation: The else part is executed if control doesn't break out of the loop.

3. What will be the output of the following Python code?

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

a) 0 1 2
b) error
c) 0 1 2 0 1 2
d) none of the mentioned

Answer: a
Explanation: The first statement creates a generator object.

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

```
x = (i for i in range(3))
for i in x:
    print(i)
for i in x:
    print(i)
```
a) 0 1 2
b) error
c) 0 1 2 0 1 2
d) none of the mentioned

Answer: a
Explanation: We can loop over a generator object only once.

5. What will be the output of the following Python code?
```
string = "my name is x"
for i in string:
    print (i, end=", ")
```
a) m, y, , n, a, m, e, , i, s, , x,
b) m, y, , n, a, m, e, , i, s, , x
c) my, name, is, x,
d) error

Answer: a
Explanation: Variable i takes the value of one character at a time.

6. What will be the output of the following Python code?
```
string = "my name is x"
for i in string.split():
    print (i, end=", ")
```
a) m, y, , n, a, m, e, , i, s, , x,
b) m, y, , n, a, m, e, , i, s, , x
c) my, name, is, x,
d) error

Answer: c
Explanation: Variable i takes the value of one word at a time.

7. What will be the output of the following Python code snippet?
```
a = [0, 1, 2, 3]
for a[-1] in a:
    print(a[-1])
```
a) 0 1 2 3
b) 0 1 2 2
c) 3 3 3 3
d) error

Answer: b
Explanation: The value of a[-1] changes in each iteration.

8. What will be the output of the following Python code snippet?

```
a = [0, 1, 2, 3]
for a[0] in a:
    print(a[0])
```

a) 0 1 2 3
b) 0 1 2 2
c) 3 3 3 3
d) error

Answer: a
Explanation: The value of a[0] changes in each iteration. Since the first value that it takes is itself, there is no visible error in the current example.

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

```
a = [0, 1, 2, 3]
i = -2
for i not in a:
    print(i)
    i += 1
```

a) -2 -1
b) 0
c) error
d) none of the mentioned

Answer: c
Explanation: SyntaxError, not in isn't allowed in for loops.

10. What will be the output of the following Python code snippet?

```
string = "my name is x"
for i in ' '.join(string.split()):
    print (i, end=", ")
```

a) m, y, , n, a, m, e, , i, s, , x,
b) m, y, , n, a, m, e, , i, s, , x
c) my, name, is, x,
d) error

Answer: a
Explanation: Variable i takes the value of one character at a time.