PYTHON OBJECTIVES ANSWERS

1. A
2. B
3. B
4. A
5. A
6. C
7. D
8. B
9. D
10. B
11. C
12. A
13. C
14. B
15. B
16. C
17. B
18. D
19. B
20. C
21. B
22. D
23. D
24. C
25. A
26. C
27. B
28. C
29. B
30. C
31. B
32. B
33. B
34. B
35. C
36. A
37. C
38. A
39. B

40. C
41. A
42. A
43. C
44. B
45. C
46. B
47. D
48. A
49. C
50. C

Explanation: Delimiters are used to separate column data for each row in the file and can be any character or character set such as the comma (,) or tab (\t).

51.A

Explanation: While not required, typically the first row in separated value files contains the name of the data columns.

52.B

Explanation: csv.reader() has a keyword argument called delimiter that is used to signal what delimiter is used for the file. By default, the delimiter value is the comma (','). You'll need to change that whenever that delimiter is different.

53. B

Explanation:When iterating over an object returned from csv.reader(), the contents returned is the row data as a list of string

For example, given a CSV file with the following contents:

```
Name,Hire Date,Salary,Sick Days remaining
Fred,10/10/10,10000,10
```

The code above would have an output like so:

```
for item in csv_reader:
print(item)
['Name', 'Hire Date', 'Salary', 'Sick Days remaining']
['Fred', '10/10/10', '10000', '10']
```

54. C,

Explanation:The field names passed into the Dict Writer constructor identify the column names for the CSV file. Therefore each key in the input dictionary passed into the method .writer now() must match up to the field names initially passed into the Dict Writer object.

The example above should be corrected to:

```
with open('test_file.csv', mode='w') as csv_file:

    writer = csv.DictWriter(
        csv_file,
        fieldnames=['first_col', 'second_col']

    writer.writeheader()

    # Notice the change in the keys below
    writer.writerow({'first_col':'value1', 'second_col':'value2'})
```

55. C,

Explanation: pandas has multiple read_functions that are used to load file data into a pandas.DataFrame. This DataFrames used to work with table like data structures in Python more easily. When working with CSV files, you should use the pandas.read_csv() function. Here's a quick example. Given the following CSV file:

```
Name,Hire Date,Salary,Sick Days remaining
Fred,10/10/10,10000,10
```

Using pandas.read_csv() would yield the following:

```
pandas.read_csv('hrdata.csv')
 Name Hire Date  Salary  Sick Days remaining
0  Fred  10/10/10   10000    10
```

56. A,

Explanation: The read_csv() function has a lot of possible parameters that can be used.Index_col can be used to define which column of the csv file should be used for the index column. This can be an integer that represents which column (0-indexed) of the header that should be used OR a string that represents which column value of the header should be used.

```
 pandas.read_csv('hrdata.csv', index_col='Name')
     Hire Date  Salary  Sick Days remaining
Name
Fred  10/10/10   10000                 10
```

```
pandas.read_csv('hrdata.csv', index_col=0)
    Hire Date  Salary  Sick Days remaining
Name
Fred  10/10/10  10000                 10
```

57. B,

Explanation: Most likely, you'll also want to use the second positional argument, mode. This argument is a string that contains multiple characters to represent how you want to open the file. The default and most common is 'r', which represents opening the file in read-only mode as a text file.

58. C

Explanation: Animals/feline, Since we're in the same location where the animals folder resides, the answer here is animals/feline.

59. A,

Explanation: While the bytes=True may appear to be correct, that's not a keyword argument for the open built in. The only way to work with byte data is to use the string value of 'b' along with the 'r' for reading or 'w' for writing.

60.A

Explanation: Animals/ursine/bears.gif, Since we're in the same location where the animals folder resides, the answer here is animals/ursine/bears.gif.

61. D,

Explanation: While the try/finally block is a valid way to ensure that the file is properly closed, it is still recommended to use the with statement instead. We recommend this because of its readability and ease of use to new users.

62.A

Explanation: /ursine/bears.gif, The character set .. is used to access a folder that is directly above the current working directory (cwd). In this case you can go up a single directory

../ursine/bears.gif

or you can go up even more:

../../Animals/ursine/bears.gif

though this isn't the most efficient way to reference the bears.gif file.

63. B,

Explanation: The .read_file_to_str() might seem like the right one, it is in fact not a method of the built in file object. The easiest option is to use the .read() method. Remember though that if you pass in a integer value, the entire file is no longer read and instead n number of bytes will be read and returned back, where n is the integer value you passed in.

64. A,

Explanation:First, move the decimal point one place to the right to get17.3.

Then round up to the nearest integer, which is18.

Finally, shift the decimal point back one place to get1.8.

65. B

66. B,

Explanation: When you truncate a positive number, you just chop off digits past the digit to which you are rounding.

For example, truncating1.7365to three decimal places results in1.736. The result is the same as rounding down to three decimal places.

On the other hand, truncating-1.7365to three decimal places results in-1.736, which is to the right of -1.7365 on the number line, and is therefore the same as rounding up to three decimal places.

67. A,

Explanation:When you "round half away to zero", you round ties up for positive numbers, and *down* for negative numbers.

Shifting the decimal place in-0.045 to the right two decimal places gives-4.5. This is exactly between-4and-5, but since we are rounding *away* from zero, we round *down* to-5.

Finally, shift the decimal point back two places to the left to get-0.05.

68. D,

Explanation: As mentioned in the Round Half to Even section of the How to Round Numbers in Python article, the round() function uses the “round half to even” strategy.

69. A,

Explanation: First, shift the decimal point one place to the right to get46.5. This is exactly half way between46 and47.

According to the “round half to even” strategy, we should round 46.5to nearest even number, which is46.

Shifting the decimal point back to the left one place gives4.6.

70 . A

Explanation: The diamond problem arises when multiple inheritance is used. This problem arises because the same name member functions get derived into a single class. Which in turn creates ambiguity in calling those methods.

71. C

Explanation: For the implementation of multiple inheritance, there must be at least 3 classes in a program. At least 2 base classes and one class to inherit those two classes. If lesser, it becomes single level inheritance.

72. B

Explanation: The constructors of parent class will be called first. In that, the constructor of the classes will be called in the same sequence as that mentioned in class definition inheritance. Since class B is mentioned first for inheritance, its constructor will be called first.

73. A

Explanation: The derived class will not be able to access any members of the base classes. Since private member’s are not inheritable. It leads to no use of multiple inheritance.

74. D

Explanation: The derived class must not be abstract. This is because the abstract classes doesn't have constructor and hence we won't be having capability to have instances. This will restrict use of multiple inheritance.

75. B

Explanation: Only if the class is being derived from other derived class, it can be called as multilevel inheritance. If a class is derived from another class, it is single level inheritance. There must be more than one level of inheritance.

76. A

Explanation: A is parent of all other classes indirectly. Since A is parent of B and B is parent of C and so on till E. Class A constructor will be called first always.

77. A

Explanation: The class with highest degree of abstraction will be the class at the 1st level. You can look at a simple example like, a CAR is more abstract than SPORTS CAR class. The level of abstraction decrease with each level as more details comes out.

78. D

Explanation: The multilevel inheritance allows any number of levels of inheritance. This is the maximum flexibility feature to make the members available to all the new classes and to add their own functionalities. The code resonbility is used too.

79. B

Explanation: Each class constructor must be called before creating object of any subclass. Hence it will be mandatory to call the constructors of parent classes explicitly with parameters. This will make all the previous class member be initialized and then the class in use will be able to create the object.

80. D

Explanation: Any type of exception can be handled by using class Exceptions. An object of this class is created which can manipulate the exception data. The data can be used to display the error or to run the program further based on error produced.

81. C

Explanation: The exceptions class is having two other derived classes which are of runtime exception handler and for other type of exceptions handling. The runtime exception handler is used to handle the exceptions produced during run time and same with case of other exceptions.

82. A

Explanation: Two blocks that are used to check for errors and to handle the errors are try and catch block. The code which might produce some exceptions is placed inside the try block and then the catch block is written to catch the error that is produced. The error message or any other processing can be done in catch block if the error is produced.

83. A

Explanation: The object must be created of a specific class of which the error has occurred. If the type of error is unknown then we can use an object of class Exceptions. This object will be able to handle any kind of exception that a program might produce.

84. C

Explanation: There is a specific class to handle each type of exceptions that might be produced in a program. The input and output exceptions can be handled by an object of class IO Excceptions. This class handles all type of input and output exceptions.

85. D

Explanation: It’s not mandatory that either base class or derived class can give rise to exceptions. The exceptions might get produced from any class. The exceptions depends on code.

86. A

Explanation: It is a condition for writing the catch blocks for base and derived classes. It is mandatory to write derived class catch block first because the errors produced by the derived class must be handled first.

87. B

Explanation: The catching of base class exception before derived class is not allowed in java. The compiler itself doesn't allow this declaration. It produces an error.

88. D

Explanation: It is the exception handler that handles the exceptions when the class used is not found in the program. This is done to handle all the undefined class exceptions. This can be due to a command line error.

89. C

Explanation: The purpose of exception handling is to handle the unexpected errors in the program. If base class might produce some error then its catch block must be given and if the derived class might produce some error then it must be given a specific catch block too.

90 . C

Explanation: The infinite loops doesn't stop running once started. There must be a way to stop the loop but that is always an improper termination. Infinite loops may keep on using more memory and hence would result in memory error.

91. B

Explanation: The functions which are made common, with respect to definition and data usage, to all the objects. These functions are able to access the static data members of a class.

92. C

Explanation: The static member functions are common for all the objects. These functions can use only the static members of a class in which those are defined. This is because other members change with respect to each object created.

93. D

Explanation: The scope resolution operator must be used to access the static member functions with class name. This indicates that the function belongs to the corresponding class.

94. D

Explanation: The static members are created only once. Then those members are reused whenever called or invoked. Memory is allocated only once.

95. B

Explanation: The static member functions can't be overloaded because the definition must be the same for all the instances of a class. If an overloaded function have many definitions, none of them can be made static.

96. D

Explanation: The static member functions can't be made const, since any object or class itself should be capable of making changes to the function. And the function must retain all changes common to all the objects.

97. C

Explanation: The single colon can't be used in any way in order to access the static members of a class. Other symbols can be used according to the code and need.

98. C

Explanation: Since the members are created once and are common for all the instances, those can be initialized inside the class. Those doesn't change with each object being created hence can be defined inside the class once for all.

99. B

Explanation: The static data members can never be mutable. There copies are not made. Since those are common and created only once.

100. A

Explanation: The static members are property of class as a whole. There is no need of specific objects to call static members. Those can be called directly or with class name.

101. C

Explanation: SciPy provides a lot of scientific routines that work on top of NumPy.

102. A

Explanation: If a dimension is given as -1 in a reshaping operation, the other dimensions are automatically calculated.

103. A

Explanation: When arrange is used with floating point arguments, it is generally not possible to predict the number of elements obtained.

104 . D

Explanation: The number of axes is called rank.

105. B

Explanation: column_stack is equivalent to v stack only for 1D arrays.

106. A

Explanation: numpy.array is not the same as the Standard Python Library class array .array.

107. A

Explanation: The copy method makes a complete copy of the array and its data.

108. B

Explanation: Length of the 1D boolean array must coincide with the length of the dimension (or axis) you want to slice.

109. A

Explanation: ndarray.data is the buffer containing the actual elements of the array.

110. D

Explanation: NumPy is the fundamental package for scientific computing with Python.

111. C

Explanation: Adjusting the size of the buffer may therefore alter the speed at which ufunc calculations of various sorts are completed.

112. B

Explanation: ufunc instances can also be produced using the from pyfunc factory function.

113. A

Explanation: Universal functions in NumPy are flexible enough to have mixed type signatures

114. B

Explanation: The optional output arguments of the function can be used to help you save memory for large calculations.

115. C

Explanation: The output of the ufunc is not necessarily an ndarray, if all input arguments are not ndarrays.

116. B

Explanation: setter sets how floating-point errors are handled.

117. B

Explanation: All ufuncs can take output arguments. If necessary, output will be cast to the data-type of the provided output array.

118. C

Explanation: fmod function return the element-wise remainder of division.

119. A

Explanation: iscomplex function returns a bool array, where True if input element is complex.

120 .A

Explanation: If the class has an __array_wrap__ method, the returned ndarray result will be passed to that method just before passing control back to the caller.

121. C

Explanation: The length of a Series cannot be changed.

122. D

Explanation: Some elements may be close to one another according to one distance and farther away according to another.

123. A

Explanation: You can read data from a CSV file using the read_csv function.

124. A

Explanation: You get columns out of a DataFrame the same way you get elements out of a dictionary.

125. C

Explanation: Panel is generally 3D labelled.

126. A

Explanation: NumPy is the fundamental package for scientific computing with Python.

127. B

Explanation: DataFrame is a container for Series, and Panel is a container for DataFrame Objects.

128. C

Explanation: Bokeh is a Python interactive visualization library for large datasets that natively uses the latest web technologies.

129. A

Explanation: It has great support for pandas data objects.

130 . A

Explanation: Time series and cross-sectional data are special cases of panel data.

131. A

Explanation: PyDatastream is a Python interface to the Thomson Dataworks Enterprise (DWE/Datastream) SOAP API to return indexed Pandas DataFrames or Panels with financial data.

132. A

Explanation: Bokeh goal is to provide elegant, concise construction of novel graphics in the style of D3.

133. A

Explanation: Geopandas extends pandas data objects to include geographic information which support geometric operations.

134. A

Explanation: If your work entails maps and geographical coordinates, and you love pandas, you should take a close look at Geopandas.

135. C

Explanation: Spyder is a cross-platform Qt-based open-source Python IDE.

136. B

Explanation: freedapi module requires a FRED API key that you can obtain for free on the FRED website.

137. B

Explanation: Spyder show both “column wise min/max and global min/max coloring.

138. A

Explanation: scikit-learn is built on NumPy, SciPy, and matplotlib.

139. B

Explanation: Seaborn has great support for pandas data objects.

140. A

Explanation: It aims to provide a pandas-like and pandas-compatible toolkit for analytics on multi- dimensional arrays.

141. B

Explanation: SparseArray is a 1-dimensional ndarray-like object storing only values distinct from the fill_value.

142.D

Explanation: The to_sparse method takes a kind argument and a fill_value.

143. D

Explanation: SparseArray can be converted back to a regular ndarray by calling to_dense.

144. A

Explanation: To create one, simply call the SparseList constructor with a fill_value.

145. A

Explanation: to_array. append can accept scalar values or any 1-dimensional sequence.

146. A

Explanation: Experimental api to transform between sparse pandas and scipy.sparse structures.

147. B

Explanation: The block format tracks only the locations and sizes of blocks of data.

148. A

Explanation: For DataFrames, likewise, in applies to the column axis.

149. B

Explanation: ix and reindex are 100% equivalent.

150. A

Explanation: This happens in an if or when using the boolean operations, and, or, or not.
