**Final Test 1**

**Question 1:** Which of the following statements are **TRUE**? (Select two answers)

A. A source file named \_\_init\_\_.py is used to mark a directory/ folder as containing a Python package, and to initialize the package.

B. The .pby extension marks files that contain Python semi-compiled byte-code.

C. A programmer is obliged to manually create a directory/ folder named \_\_pycache\_\_ inside every package.

D. The variable named \_\_name\_\_ is a string containing the module name.

**Question 2:** What is **TRUE** about the built-in dir() mechanism in the context of modules and packages?

A. It is a dictionary contained by a module reflecting the module contents. B. It is a list contained by a module reflecting the module contents.

C. It is a function which can be invoked with a module passed as an argument in order to obtain the module content.

D. It is a method which can be invoked from within a module in order to obtain the module contents.

**Question 3:** A function named f() is included in a module named m, and the module is a part of a package named p. Which of the following code snippets allows you to properly invoke the function? (Select two answers)

A. from p.m import f

f()

B. import p.m.f

f()

C. import p.m

p.m.f()

D. import p

m.f()

**Question 4:** Which of the following functions come from the math module? (Select two answers)

A. seed()

B. processor()

C. sqrt()

D. hypot()

**Question 5:** What is the expected output of the following code?

>>> import math

>>> x = -1.5

>>> print(abs(math.floor(x) + math.ceil(x)))

A. 2

B. 3

C. -2

D. -3

**Question 6:** Which of the following variables will Python consider to be **private**?

A. private\_\_data

B. \_\_privatedata

C. privatedata\_\_

D. \_privatedata\_

**Question 7:** What is the expected output of the following code?

>>> consts = (3.141592, 2.718282)

>>> try:

>>> print(consts[2])

>>> except Exception as exception:

>>> print(exception.args)

>>> else:

>>> print(“(‘success’)”

A. 2.718282

B. (‘tuple index out of range’,)

C. 3.141592

D. (‘success’)

**Question 8:** What is the expected output of the following code?

>>> def fun(x):

>>> assert x >= 0

>>> return x \*\* 0.5

>>> def mid\_level(x):

>>> try:

>>> fun(x)

>>> except Error:

>>> raise

>>> try:

>>> x = mid\_level(-1)

>>> except RuntimeError:

>>> x = -1

>>> except:

>>> x = -2

>>> print(x)

A. An error message appears on the screen

B. 0

C. -2

D. -1

**Question 9:** Which of the following are the names of built-in Python exceptions? (Select two answers).

A. KeyError

B. AssertionError

C. LookupException

D. ProgramTooComplicatedError

**Question 10:** What is expected output of the following code?

>>> x, y = 3.0, 0.0

>>> try:

>>> z = x/y

>>> except ArithmeticError:

>>> x = -1

>>> else:

>>> z = -2

>>> print(z)

A. An error message appears on the screen

B. +INF

C. -2

D. -1

**Question 11:** Which of the following messages will appear on the screen when the code is run? (Select two answers)

>>> class Accident (Exception):

>>> def \_\_init\_\_(self, message):

>>> self.message = message

>>> def \_\_str\_\_(self):

>>> return “problem”

>>> try:

>>> print(“action”)

>>> raise Accident (“accident”)

>>> except Accident as accident:

>>> print(accident)

>>> else:

>>> print(“success”)

A. action

B. accident

C. problem

D. success

**Question 12:** Which of the following are valid Python string literals? (Select two answers)

A. “King’s Cross Station”

B. “/“

C. ‘All the king’s horses”

D. “””The Knights Who Say ‘Ni!’”””

**Question 13:** Which of the following snippets can be used to build a new string consisting of sorted characters contained in the ‘zyx’ string assigned to the letters variables? (Select two answers)

>>> letters = ‘zyx’

A. new\_string = sorted(letters)

B. new\_string = ‘ ‘.join(sorted(letters))

C. tmp = letters.sort()

new\_string = str(tmp)

D. tmp = list(letters)

tmp.sort()

new\_string = ‘ ‘.join(tmp)

**Question 14:** Which of the following assignments can be performed without raising any exceptions? (Select two answers)

A. s = ‘rhyme’

s = s[-2]

B. s = ‘rhyme’

s = s[::2]

C. s = ‘rhyme’

s = s[9]

D. s = ‘rhyme’

s[0] = s[1]

**Question 15:** What is expected output of the following code?

>>> plane = “Cessna”

>>> counter = 0

>>> for c in plane \* 2:

>>> if c in [“e”, “a”]:

>>> counter +=1

>>> print(counter)

A. 0

B. 2

C. 4

D. The code is erroneous and cannot be run

**Question 16:** What is the expected output of the following code?

>>> foo = “Mary had 21 little sheep”

>>> print(foo.split()[2].isdigit())

A. 2

B. 21

C. True

D. False

**Question 17:** Which of the following expressions evaluate to **TRUE** and raise no exception? (Select two answers)

A. str(None) != “None”

B. str(None) == None

C. ‘ ‘ \* 0 < 1 \* ‘ ‘

D. ‘Analog’ < ‘analog’

**Question 18:** Which of the following expressions evaluate to **TRUE** and raise no exception? (Select two answers)

A. ‘bc’ in ‘abc’

B. ‘ ‘ not in ‘ ‘

C. ‘xyz’ not in ‘uvwxyz’

D. ‘ ‘ in ‘alphabet’

**Question 19:** Which of the following are character encoding standard names? (Select two answers)

A. ASCII

B. Unicode

C. Intcod

D. Unilang

**Question 20:** Given the code below, which of the expressions will evaluate to **TRUE**? (Select two answers)

>>> class Top:

>>> value = 3

>>> def say(self):

>>> return self.value

>>> class Middle(Top):

>>> value = 2

>>> class Bottom(Middle):

>>> def say(self):

>>> return -self.value

>>> short = Bottom()

>>> tall = Top()

>>> average = Middle()

A. tall.say() = 2

B. average.value == 2

C. isinstance(average, Bottom)

D. short.value == 2

**Question 21:** What is the expected behavior of the following snippet?

>>> class Team:

>>> def show\_ID(self):

>>> print(self.get\_ID())

>>> def get\_ID(self):

>>> return “anonymous”

>>> class A(Team):

>>> def get\_ID(self):

>>> return “Alpha”

>>> a = A()

>>> a.show\_ID()

A. It raises an exception.

B. It outputs anonymous.

C. It outputs an empty line.

D. It outputs Alpha.

**Question 22:** Given the code below, indicate the code lines which correctly increment the \_\_element variable by one. (Select two answers)

>>> class BluePrint:

>>> \_\_element = 1

>>> def \_\_init\_\_(self):

>>> self.component = 1

>>> def \_\_action(self):

>>> pass

>>> product = BluePrint()

A. \_product\_\_element += 1

B. product.\_\_BluePrint\_\_element += 1

C. BluePrint.element += 1

D. BluePrint.\_\_BluePrint\_\_element += 1

**Question 23:** Given the code below, which of the following expressions will evaluate to **TRUE**? (Select two answers)

>>> class Alpha:

>>> value = “Alpha”

>>> def say(self):

>>> return self.value.lower()

>>> class Beta(Alpha):

>>> value = “Beta”

>>> class Gamma (Alpha):

>>> def say(self):

>>> return self.value.upper()

>>> class Delta(Gamma, Beta):

>>> pass

>>> d = Delta()

>>> b = Beta()

A. Alpha in Delta.\_\_bases\_\_

B. d.value == “Alpha”

C. d.say() = “BETA”

D. isinstance(d, Beta)

**Question 24:** Assuming that the following code has been executed successfully, indicate the expressions which evaluate to **TRUE** and don’t raise any exceptions. (Select two answers)

>>> class Class:

>>> class\_var = 1

>>> def \_\_init\_\_(self):

>>> self.insintance\_var = 1

>>> def method(self):

>>> pass

object = Class()

A. Class.\_\_dict\_\_[‘method’] != None

B. len(object.\_\_dict\_\_) == len(Class.\_\_dict\_\_)

C. ‘\_\_dict\_\_’ in Class.\_\_dict\_\_

D. object.\_\_dict\_\_[‘method’] != None

**Question 25:** What is the expected output of the following code?

>>> class Ceil:

>>> Token = 1

>>> def get\_token(self):

>>> return 1

>>> class Floor(Ceil):

>>> def get\_token(self):

>>> return 2

>>> def set\_token(self):

>>> pass

>>> holder = Floor()

>>> print(hasattr(holder, “Token”), hasattr(Ceil, “set\_token”))

A. True False

B. False False

C. True True

D. False True

**Question 26:** What is expected behavior of the following code?

>>> class Tin:

>>> label = “Soup”

>>> def \_\_init\_\_(self, prefix):

>>> self.name = prefix + “ “ + Tin.label

>>> can\_1 = Tin(“Tomato”)

>>> can\_2 = Tin(“Chicken”)

>>> print(can\_1.label == can\_2.label)

A. It outputs True.

B. The code is erroneous and it will raise an exception.

C. It outputs False.

D. It outputs None.

**Question 27:** Which of the following classes have valid constructors? (Select two answers)

A. Class Gimel:

def \_\_init\_\_():

self.attribute = True

B. Class Dalet:

def \_\_init\_\_(self):

return False

C. class Bet:

def \_\_init\_\_(self):

raise ArithmeticError

D. class Aleph:

def \_\_init\_\_(self):

self.attribute = True

**Question 28:** What is the expected output of the following code?

>>> class Top:

>>> def \_\_str\_\_(self):

>>> return ‘1’

>>> class Left(Top):

>>> def \_\_str\_\_(self):

>>> return ‘2’

>>> class Right(Top):

>>> def \_\_str\_\_(self):

>>> return ‘3’

>>> class Bottom(Right, Left):

>>> pass

>>> object = Bottom()

>>> print(object)

A. 1

B. 2

C. 3

D. An empty line.

**Question 29:** What is **TRUE** about object-oriented programming (OOP)? (Select two answers)

A. A class is like a blueprint used to construct object.

B. A class may exist without its objects, while objects cannot exist without their class.

C. *Polymorphism* is a phenomenon which allows you to have many classes of the same name.

D. A relation between a superclass and its subclass is known as *fraternity*.

**Question 30:** Given the class below, indicate a method which will correctly provide the value of the rack field?

>>> class Storage:

>>> def \_\_init\_\_(self):

>>> self.rack = 1

#insert a method here

>>> stuff = Storage()

print(stuff.get())

A. def get():

return rack

B. def get(self):

return self.rack

C. def get():

return self.rack

D. def get(self):

return rack

**Question 31:** If you want to check if a Python file is either used as a module or run as a standalone program, you should check a built-in variable named:

A. \_\_run\_mode\_\_

B. \_\_used\_as\_\_

C. \_\_module\_name\_\_

D. \_\_name\_\_

Question 32: Which of the following statements are TRUE? (Select two answers)

A. Trying to write a file opened in read-only mode removes its contents.

B. The second argument of the open() function is a string.

C. Read, write, and delete are the names of file open modes.

D. The open() function raises an exception when its operation fails.

**Question 33:** What is the expected output of the following code?

>>> def quote(quo):

>>> def embed(str):

>>> return quo + str + quo

>>> return embed

>>> dblq = quote(‘ “ ‘)

>>> print(dblq(‘Jane Doe’))

A. “Jane Doe”

B. “‘Jane Doe’”

C. Jane Doe

D. ‘Jane Doe’

**Question 34:** What is the expected output of the following code if the file named existing\_text\_file is a non-zero length text file located in the working directory, and the open() function invocation is successful?

>>> try:

>>> f = open(“existing\_text\_file”, “rt”)

>>> spam = f.readlines()

>>> print(len(spam))

>>> f.close()

>>> except IOError:

>>> print(-1)

A. The number of lines contained inside the file.

B. The length of the last line from the file.

C. -1

D. The length of the first line from the file.

**Question 35:** Which of the following lines contain valid Python code? (Select two answers)

A. lambda x, y -> x \*\* y

B. lambda x, y: ‘0123456789’ [x:y]

C. lambda f(x, y): return x >> y

D. lambda x, y: x + y

**Question 36:** Which method is used to break the connection between the file handle and a physical file?

A. lock()

B. close()

C. shutup()

D. disconnect()

**Question 37:** What is the expected output of the following code?

>>> vect = [“alpha”, “bravo”, “charlie”]

>>> new\_vect = filter(lambda s: s[-1].upper() in [“A”, “O”], vect) >>> for x in new\_vect:

>>> print(x[1], end = “ “)

A. RH

B. lr

C. rh

D. LR

**Question 38:** What is expected output of the following code?

>>> l = [x for x in range(1, 10, 3) if x % 2 == 0]

>>> print(len(l))

A. 2

B. 8

C. 4

D. 1

**Question 39:** What is expected output of the following code?

>>> v = [1, 2, 3]

>>> def g(a, b, m):

>>> return m(a, b)

>>> print(g(1, 1, lambda x, y: v[ x : y + 1]))

A. [2]

B. []

C. [3]

D. [1]

**Question 40:** What is the expected output of the following code?

>>> def f(1):

>>> return 1(-1, 3)

>>> print(f(lambda x, y: x if x > y else y))

A. -3

B. 0

C. None

D. 3

Question 2: AC

Question 17:B

Question 24:D