**NOTES FOR TRANSLATOR**

**Please write the translation directly at the end of each section (see note after section 1.1)**

**Please keep the italics and underlining the same (they indicate the correct answer(s))**

**Please keep code the same and do not translate**

**Do not worry about other formatting. (Each question and its separate answers are copied and pasted individually by us into another program. This is the reason these questions in this document are not identically formatted since they didn’t come from the same standard document.)**

**Self-Check Questions**

**1.1**

1. List three reasons why Python is a popular language among educators and learners.

*Python is free, and a variety of supporting tools, modules, and libraries are available at no cost to aspiring developers.*

*Python’s syntax is concise compared to that of many other languages, which means you can do more with less, reducing the demand on the programmer.*

*Python code is also easier to read than code written in many other languages because one of the central concepts in the creation of Python was that the code should resemble everyday English.*

*Python has a very active developer community that creates resources for entry-level and expert Python developers.*

*Developers can use Python to build applications using procedural, object-oriented, or functional paradigms.*

*Because Python is an interpreted language, developers can get immediate feedback without having to wait for an application to compile.*

1. List three reasons why Python is a popular language among professionals.

*Despite being relatively easy to use and read, Python is extremely robust and powerful.*

*Because Python is so easy to use and such a concise language, it is a popular choice for quickly building working prototypes.*

*Python is the language of choice for data science and is heavily used in artificial intelligence.*

*Because of its popularity in data science and AI, there is a huge ecosystem of libraries available for Python development in areas such as mathematics, statistics, machine learning, and deep learning.*

*There are exceptional libraries and frameworks to facilitate web development in Python (Django and Flask are great examples).*

*Python is already extremely popular among educators, learners, and practitioners, and the trends show that this momentum is only increasing. That means more jobs at higher salaries available to Python developers. Learning Python can help you pay the bills!*

* 1. Translation

1. Nennen Sie drei Gründe dafür, dass Python bei Lehrkräften und Studierenden beliebt ist.

*Python und viele dazugehörige Tools, Module und Bibliotheken sind kostenlos erhältlich.*

*Verglichen mit vielen anderen Programmiersprachen ist die Syntax von Python knapper und präziser. Dadurch lassen sich mehr Funktionen mit weniger Zeilen realisieren, was das Programmieren spürbar vereinfacht.*

*Python-Code wurde speziell so konzipiert, dass er dem Alltagsenglisch ähnelt. Daher lässt er sich leichter lesen als Programme in vielen anderen Sprachen.*

*Die umtriebige Python-Community stellt vielfältige Ressourcen für angehende und erfahrene Entwickler:innen bereit.*

*Python unterstützt prozedurale, objektorientierte und funktionale Programmierparadigmen.*

*Da Python eine interpretierte Programmiersprache ist, können Neuentwicklungen sofort getestet werden, ohne dass die betreffende Anwendung zuerst kompiliert werden muss.*

1. Nennen Sie drei Gründe dafür, dass Python bei Fachleuten aus dem Bereich Softwareentwicklung beliebt ist.

*Python-Code lässt sich nicht nur relativ einfach schreiben und lesen, sondern ist außerdem extrem robust und leistungsstark.*

*Als benutzerfreundliche, kompakte Programmiersprache ist Python eine naheliegende Option für die schnelle Erstellung funktionsfähiger Prototypen.*

*Python ist die Sprache der Wahl im Bereich Data Science und kommt darüber hinaus bei vielen Projekten rund um die künstliche Intelligenz zum Einsatz.*

*Wegen der weiten Verbreitung der Sprache in den Bereichen Data Science und KI sind zahlreiche Python-Bibliotheken für mathematische und statistische Anwendungen sowie Deep Learning und andere Varianten des maschinellen Lernens verfügbar.*

*Es gibt außergewöhnlich leistungsstarke Bibliotheken und Frameworks für die Entwicklung von Webanwendungen in Python (wie beispielsweise Django und Flask).*

*Da sich Python sowohl in der akademischen Lehre als auch unter den in der Branche tätigen Fachleuten einer wachsenden Beliebtheit erfreut, steigt die Zahl gut entlohnter Stellen für Python-Entwickler:innen. Somit lohnt sich das Erlernen der Programmiersprache auch unter finanziellen Aspekten!*

**1.2**

1. What is Anaconda?

a) Anaconda is a smaller, lighter version of Python used to develop mobile applications.

b) Anaconda is a nickname for Python version 3.7. Version 2.7 also has a nickname: Boa.

c) *Anaconda is a distribution that includes Python and various other tools including SciPy, NumPy, and pandas.*

1. What is NumPy?

a) a*n extremely efficient library used for computation on large data sets and multi-dimensional arrays*

b) a library used for scientific computing including linear algebra, interpolation, and signal and image processing

c) a library used for data manipulation in numerical tables and time series data

d) a tool used for numbers in Python without which you cannot use numbers in Python programming

1. What is SciPy?

a) an extremely efficient library used for computation on large data sets and multi-dimensional arrays

b) *a library used for scientific computing including linear algebra, interpolation, and signal and image processing*

c) a library used for data manipulation in numerical tables and time series data

d) science, but for Python, without which you cannot develop scientific applications

1. What is pandas?

a) an extremely efficient library used for computation on large data sets and multidimensional arrays

b) a library used for scientific computing including linear algebra, interpolation, and signal and image processing

c) *a library used for data manipulation in numerical tables and time series data*

d) a library with panda bear images, useful in clip art when looking for images of black and white, soft, cuddly mammals

**1.2 Translation**

1. Was ist Anaconda?

a) Anaconda ist eine schlankere Python-Version für die Entwicklung mobiler Apps.

b) Anaconda ist der Codename für die Python-Version 3.7. Version 2.7 trug den Codenamen Boa.

c) *Anaconda ist eine Python-Distribution, die verschiedene zusätzliche Tools wie SciPy, NumPy und Pandas enthält.*

1. Was ist NumPy?

a) *eine extrem effiziente Bibliothek für die Handhabung großer Datensätze und mehrdimensionaler Arrays*

b) eine für wissenschaftliche Berechnungen konzipierte Bibliothek mit Funktionen für lineare Algebra, Interpolation und die Signal- und Bildverarbeitung

c) eine Bibliothek mit Funktionen für die Verwaltung und Analyse von numerischen Tabellen und Zeitreihendaten

d) ein Tool zur Implementierung von Zahlen – ohne NumPy können keine Zahlen in Python-Programmen verwendet werden.

1. Was ist SciPy?

a) eine extrem effiziente Bibliothek für die Handhabung großer Datensätze und mehrdimensionaler Arrays

b) *eine für wissenschaftliche Berechnungen konzipierte Bibliothek mit Funktionen für lineare Algebra, Interpolation und die Signal- und Bildverarbeitung*

c) eine Bibliothek mit Funktionen für die Verwaltung und Analyse von numerischen Tabellen und Zeitreihendaten

d) ein Wissenschaftstool, das eine notwendige Voraussetzung für die Erstellung wissenschaftlicher Anwendungen in Python ist

1. Was ist Pandas?

a) eine extrem effiziente Bibliothek für die Handhabung großer Datensätze und mehrdimensionaler Arrays

b) eine für wissenschaftliche Berechnungen konzipierte Bibliothek mit Funktionen für lineare Algebra, Interpolation und die Signal- und Bildverarbeitung

c) *eine Bibliothek mit Funktionen für die Verwaltung und Analyse von numerischen Tabellen und Zeitreihendaten*

d) eine Clipart-Bibliothek mit Bildern von knuffigen, schwarzweiß gefärbten Pandabären

**1.3**

1. What is the Python interpreter?

a) a person employed by Python.org who is available 24/7 to interpret code

b) a compiler used to turn Python code to machine code

c) *a program that reads Python code, translates the code into something called byte code, and then executes the code in a Python Virtual machine, returning the output to the user*

d) an extension of the Jupyter Project that interprets Python code from various international languages into everyday English

1. Which of the following is a simple Hello World application in Python?

a) Hello.Python.World();

b) Console.WriteLine("Hello, World!");

c) system.out("Hello, World!");

d) *print("Hello, World!")*

1. Which of the following is a difference between interpreted languages and compiled languages? (select all that apply)

a) *Interpreted languages are parsed and executed without storing a separate machine code executable.*

b) Interpreted languages are processed into machine code and stored as separate executables for users to run.

c) *Compiled languages are processed into machine code and stored as separate executables for users to run.*

d) Compiled languages are parsed and executed without storing a separate machine code executable.

1. What is the relationship of Jupyter to IPython?

a) Jupyter and IPython are exactly the same.

b) Jupyter and IPython have nothing in common.

c) *Jupyter evolved from IPython.*

d) IPython evolved from Jupyter.

1. What is JupyterLab?

a) a console prompt that allows for better debugging and enhanced developer help

b) *an integrated development environment that allows developers to organize and develop in multiple documents with code, Markdown, and text alongside terminals and other components*

c) a high-tech, Python-based astronomical array setup to study the surface of Jupiter

d) a synonym for Python that’s friendlier for people who dislike snakes

**1.3 Translation**

1. Was ist der Python-Interpreter?

a) ein Angestellter von Python.org, der rund um die Uhr Programmcode interpretiert

b) ein Compiler, der Python-Code in Maschinensprache übersetzt

*c) ein Programm, das eingegebenen Python-Code zunächst liest und in sogenannten Bytecode übersetzt und dann sowohl die Ausführung in einer Python Virtual Machine veranlasst als auch die Ausgabe des Outputs übernimmt*

d) eine Erweiterung des Jupyter-Projekts, das Python-Code aus verschiedenen Sprachregionen in die englische Alltagssprache übersetzt

1. Bei welcher der folgenden Codezeilen handelt es sich um ein einfaches Hallo-Welt-Programm in Python?

a) Hello.Python.World();

b) Console.WriteLine("Hello, World!");

c) system.out("Hello, World!");

d) *print("Hello, World!")*

1. Welche der folgenden Aussagen zu interpretierten und kompilierten Programmiersprachen sind zutreffend?

a) *Programme in interpretierten Sprachen werden geparst und ausgeführt,* *ohne dass dafür eine ausführbare Datei in Maschinensprache erstellt werden muss.*

b) Programme in interpretierten Sprachen werden in Maschinencode übersetzt und auf den Endgeräten als ausführbare Dateien gespeichert.

c) *Programme in kompilierten Sprachen werden in Maschinencode übersetzt und auf den Endgeräten als ausführbare Dateien gespeichert.*

d) Programme in kompilierten Sprachen werden geparst und ausgeführt, ohne dass dafür eine ausführbare Datei in Maschinensprache erstellt werden muss.

1. In welcher Beziehung stehen Jupyter und IPython?

a) Jupyter und IPython sind miteinander identisch.

b) Jupyter und IPython haben nichts gemeinsam.

c) *Jupyter ist aus IPython entstanden.*

d) IPython ist aus Jupyter entstanden.

1. Was ist JupyterLab?

a) eine Konsole mit erweiterten Debugging- und Hilfefunktionen für Entwicklungsteams

b) *eine integrierte Entwicklungsumgebung, die die Verwendung von Code, Markdown-Formatierungen, Text und anderen Komponenten in ein und demselben Dokument unterstützt*

c) eine mit Python-Systemen ausgestattete astronomische Beobachtungsstation, die die Oberfläche des Jupiter untersucht

d) eine alternative Bezeichnung für Python, die vor allem von Menschen mit einer Schlangenphobie bevorzugt wird

**2.1**

1. I want to create a variable to hold the value of a player’s age. Which of the potential names below are valid? (select all)

a) *&player\_age*

b) Playerage

c) Player\_age

d) \_player\_age

e) player\_age

2. I want to create a variable to hold the value of a player’s age. Which of the potential names below are valid AND conventional? (select all)

a) &player\_age

b) Playerage

c) Player\_age

d) \_player\_age

e) *player\_age*

3. Which of the following will create a variable representing a birth year and give it a value of 2000 in a way that is both valid and conventional in Python?

a) Birth year = 2000

b) my.birth.year = 2000

c) 2000 = birth year

d) *birth\_year = 2000*

4. Given what we’ve seen so far in Python code, when the interpreter gives you an error of “invalid syntax,” what is most likely the cause?

a) The Python interpreter is malfunctioning.

b) *I tried to use a variable name that is not valid.*

c) I have a literal on the left-hand side of the assignment operator.

d) I misspelled a variable name.

5. What is the value of player\_weight at the end of this code:

player\_weight = 73  
player\_weight = player\_weight + 3  
player\_weight = 80

player\_weight = player\_weight - player\_weight

a) 73

b) 76

c) 80

d) 67

e) *0*

**2.1 Translation**

1. Sie möchten eine Variable zur Speicherung des Alters eines Fußballspielers/einer Fußballspielerin erstellen. Welche der folgenden Bezeichnungen sind in Python gültig?

a) &player\_age

b) ***Playerage***

c) ***Player\_age***

d) ***\_player\_age***

e) ***player\_age***

2. Sie möchten eine Variable zur Speicherung des Alters eines Fußballspielers/einer Fußballspielerin erstellen. Welche der folgenden Bezeichnungen sind gültig UND konventionell?

a) &player\_age

b) Playerage

c) Player\_age

d) \_player\_age

e) ***player\_age***

3. Durch welche der folgenden Codezeilen wird in Python einer sowohl gültigen als auch konventionellen Variable zur Speicherung des Geburtsjahres der Wert 2000 zugewiesen?

a) Birth year = 2000

b) my.birth.year = 2000

c) 2000 = birth year

d) ***birth\_year = 2000***

4. Bei Ihren ersten Programmierversuchen gibt der Python-Interpreter die Fehlermeldung „invalid syntax“ aus. Was kann – nach Ihrem bisherigen Wissen über die Programmiersprache – die Ursache dafür sein?

a) Der Python-Interpreter hat einen Funktionsfehler.

b) ***Sie haben versucht, einen ungültigen Variablennamen zu verwenden.***

c) Sie haben ein Literal auf der linken Seite des Zuweisungsoperators eingegeben.

d) Sie haben einen Variablennamen falsch geschrieben.

5. Welchen Wert hat die Variable „player\_weight“ nach der Ausführung der folgenden Code-Zeilen:

player\_weight = 73  
player\_weight = player\_weight + 3  
player\_weight = 80

player\_weight = player\_weight - player\_weight

a) 73

b) 76

c) 80

d) 67

e) ***0***

**2.2**

1. What type of value is 0o42?

a) int

b) float

c) complex

d) *octal*

e) hexadecimal

1. What type of value is 42?

a) *int*

b) float

c) complex

d) octal

e) hexadecimal

1. What type of value is 4 + 2J?

a) int

b) float

c) *complex*

d) octal

e) hexadecimal

1. What type of value is 4.2?

a) int

b) *float*

c) complex

d) octal

e) hexadecimal

1. What is the result of this equation: int(5.8 \* 2)?

a) 0

b) 10

c) *11*

d) 12

e) a syntax error

**2.2 Translation**

1. Zu welchem Datentyp gehört der Wert 0o42?

a) Ganzzahl (int)

b) Gleitkommazahl (float)

c) komplexe Zahl

d) *Oktalzahl*

e) Hexadezimalzahl

1. Zu welchem Datentyp gehört der Wert 42?

a) *Ganzzahl (int)*

b) Gleitkommazahl (float)

c) komplexe Zahl

d) Oktalzahl

e) Hexadezimalzahl

1. Zu welchem Datentyp gehört der Wert 4 + 2J?

a) Ganzzahl (int)

b) Gleitkommazahl (float)

c) *komplexe Zahl*

d) Oktalzahl

e) Hexadezimalzahl

1. Zu welchem Datentyp gehört der Wert 4.2?

a) Ganzzahl (int)

b) *Gleitkommazahl (float)*

c) komplexe Zahl

d) Oktalzahl

e) Hexadezimalzahl

1. Nennen Sie das Ergebnis der folgenden Rechenoperation: int(5.8 \* 2)

a) 0

b) 10

c) *11*

d) 12

e) Syntaxfehler

**2.3**

1. What is the purpose of an escape character?

a) to escape out of the Python prompt and go back to the main menu

b) to use triple quotes effectively, (""") or (''')

c) *to include special characters in a string like a backslash or quotation mark*

d) to add two strings together

1. How would you put quotation marks into a string in Python?

a) *my\_string = "this has \"quotes\"."*

b) my\_string = "this has """quotes"""."

c) my\_string = "this has \t"quotes\t"."

d) my\_string = "this has " + "quotes" + "."

1. How do you get the length of a string in Python?

a) String.get\_length(my\_string)

b) my\_string.len

c) my\_string + length

d) *len(my\_string)*

1. If I have a string “my\_string“ with the value “I love Python!,” how do I substring to get just “Python!”?

a) *my\_string[7:]*

b) my\_string.Substring(7, 13)

c) my\_string.Substring(7, 6)

d) my\_string[7::2]

1. If I have a string “my\_string” with the value “I love Python!,” how do I substring to get just “love”?

a) my\_string[2:4]

b) my\_string.Substring(2, 6)

c) my\_string.Substring(2, 4)

d) *my\_string[2:6]*

**2.3 Translation**

1. Durch die Eingabe eines Escape-Zeichens können Sie ...

a) ... die Python-Eingabeaufforderung verlassen und zum Hauptmenü zurückkehren.

b) ... die Nutzung von dreifachen Anführungszeichen (""" oder ''') effektiver gestalten.

c) *... einen Backslash, ein Anführungszeichen oder ein anderes Sonderzeichen in einen String einfügen.*

d) ... zwei Strings miteinander verbinden.

1. Wie können Sie Anführungszeichen in einen Python-String aufnehmen?

a) ***my\_string = "this has \"quotes\"."***

b) my\_string = "this has """quotes"""."

c) my\_string = "this has \t"quotes\t"."

d) my\_string = "this has " + "quotes" + "."

1. Welche Codezeile liefert Ihnen in Python die Länge eines Strings?

a) String.get\_length(my\_string)

b) my\_string.len

c) my\_string + length

d) ***len(my\_string)***

1. Sie haben eine String-Variable mit der Bezeichnung „my\_string“ und dem Wert „I love Python!“ erstellt. Welche der folgenden Codezeilen liest aus dieser Variable lediglich den Teilstring „Python!“ aus?

a) ***my\_string[7:]***

b) my\_string.Substring(7, 13)

c) my\_string.Substring(7, 6)

d) my\_string[7::2]

1. Sie haben eine String-Variable mit der Bezeichnung „my\_string“ und dem Wert „I love Python!“ erstellt. Welche der folgenden Codezeilen liest aus dieser Variable lediglich den Teilstring „love“ aus?

a) my\_string[2:4]

b) my\_string.Substring(2, 6)

c) my\_string.Substring(2, 4)

d) ***my\_string[2:6]***

**2.4**

1. What is the purpose of a collection in Python?

a) to allow you to organize data more effectively

b) to let the programmer combine multiple pieces of data into one variable object

c) to simplify code

d) *all of the above*

1. What is the proper way to add the number 24 as an element to the set that I’ve named “my\_favorite\_numbers”?

*my\_favorite\_numbers.add(24)*

1. What is the proper way to add the number 24 at index 3 of the list that I’ve named “my\_favorite\_numbers”?

*my\_favorite\_numbers.insert(3, 24)*

1. What is a key difference between a dictionary and a set?

a) Dictionaries and sets are the same thing.

b) *Dictionaries store elements in key/value pairs, and sets do not.*

c) Sets store elements in key/value pairs, and dictionaries do not.

d) Dictionaries and sets are nothing like each other.

**2.4 Translation**

1. Weshalb bietet Python die Möglichkeit zur Einrichtung und Nutzung von Collections?

a) damit Sie Ihre Daten effektiver verwalten können

b) damit Programierer:innen verschiedene Arten von Daten in ein und demselben Variablenobjekt speichern können

c) um den Programmcode zu vereinfachen

d) *aus allen genannten Gründen (a, b und c)*

1. Sie haben ein Set mit der Bezeichnung „my\_favorite\_numbers“ erstellt. Wie können Sie diesem die Zahl 24 als zusätzliches Element hinzufügen?

*my\_favorite\_numbers.add(24)*

1. Sie haben eine Liste mit der Bezeichnung „my\_favorite\_numbers“ erstellt. Wie können Sie dieser die Zahl 24 als zusätzliches Element mit dem Index 3 hinzufügen?

*my\_favorite\_numbers.insert(3, 24)*

1. Welche der folgenden Aussagen über Dictionarys und Sets ist zutreffend?

a) Dictionarys und Sets sind ein und dasselbe.

b) *Dictionarys speichern Elemente als Schlüssel-Wert-Paare. Dies trifft auf Sets nicht zu.*

c) Sets speichern Elemente als Schlüssel-Wert-Paare. Das trifft auf Dictionarys nicht zu.

d) Dictionarys und Sets sind zwei vollkommen verschiedene Dinge.

**2.5**

1. What is the command to open a file in Python?

a) create

b) get\_file

c) file

d) *open*

1. What does “x” do when used with the file open method?

a) creates a new file, even if one exists already

b) deletes the specified file

c) *creates a new file with the specified file name; if it already exists, you will get an error message*

d) opens an existing file in append mode

1. What does “r” do when used with the file open method?

a) removes the file (deletes it)

b) opens the file in append mode

c) *opens the file in read mode*

d) opens the file but removes all content previously in the file

**2.5 Translation**

1. Mit welchem Python-Befehl öffnen Sie eine Datei?

a) create

b) get\_file

c) file

d) *open*

1. Welche Wirkung erzielen Sie, wenn Sie den Parameter "x" in der open-Methode angeben?

a) Erstellung einer neuen Datei, auch wenn der angegebene Dateiname bereits vergeben ist

b) Löschung der angegebenen Datei

c) *Erstellung einer neuen Datei mit dem angegebenen Namen; falls die betreffende Datei bereits existiert, wird eine Fehlermeldung ausgegeben*

d) Öffnung einer bestehenden Datei im Anfügemodus

1. Welche Wirkung erzielen Sie, wenn Sie den Parameter "r" in der open-Methode angeben?

a) Löschung der angegebenen Datei

b) Öffnung der angegebenen Datei im Anfügemodus

*c) Öffnung der angegebenen Datei im Lesemodus*

d) Öffnung eine bestehenden Datei – bei gleichzeitiger Entfernung aller vorher vorhandenen Inhalte

**3.1**

What is the value of my\_variable after running the following line of code in Python?

my\_variable = 2 \* (3 \*\* 2) / 4 + (9 – 3)

a) *10.5*

b) 11

c) 10

d) 11.5

2. What is the fastest way to initialize four variables (a, b, c, and d) all equal to 5?

a) variables.set(5, a, b, c, d)

b) Not possible; you can only set one variable to 5 per program

c) 5 = a, b, c, d

d) a, b, c, d = 5

e) *a = b = c = d = 5*

3. Which of the following lines of code will print the output of all four variables (a, b, c, and d)?

a) print(all)

b) print(a + b + c + d)

c) *print(a, b, c, d)*

d) print(a through d)

1. I have a variable called my\_int that holds the integer value 42. I want to convert that value to a string data type and assign that value to a variable called my\_str. Which of the following will do that?

a) my\_int = my\_str(42)

b) *my\_str = str(my\_int)*

c) my\_str = "42"

d) my\_str = my\_int.makestring( )

**3.1 Translation**

Welchen Wert hat die Variable *my\_variable*, nachdem Sie in Python die folgende Codezeile ausgeführt haben?

my\_variable = 2 \* (3 \*\* 2) / 4 + (9 – 3)

a) ***10.5***

b) 11

c) 10

d) 11.5

2. Mit welcher der folgenden Zeilen können Sie den vier Variablen a, b, c und d den Wert 5 zuweisen?

a) variables.set(5, a, b, c, d)

b) Das ist nicht möglich, da in jeder Zeile nur eine Variable mit einem Wert versehen werden kann.

c) 5 = a, b, c, d

d) a, b, c, d = 5

e) ***a = b = c = d = 5***

3. Mit welcher der folgenden Codezeilen können Sie veranlassen, dass die Werte der Variablen a, b, c und d auf dem Bildschirm ausgegeben werden?

a) print(all)

b) print(a + b + c + d)

c) ***print(a, b, c, d)***

d) print(a through d)

1. Sie haben die Ganzzahl 42 in einer Variable mit der Bezeichnung my\_int gespeichert. Nun möchten Sie diesen Wert in einen String umwandeln und ihn einer anderen Variablen namens my\_str zuweisen. Mit welcher der folgenden Zeilen können Sie dies bewerkstelligen?

a) my\_int = my\_str(42)

b) ***my\_str = str(my\_int)***

c) my\_str = "42"

d) my\_str = my\_int.makestring( )

**3.2**

1. What is the difference between an expression and a statement in Python?

a) There is no difference, other than spelling

b) *A statement is a line of code the interpreter can execute, an expression is a section of code that the interpreter evaluates to a value.*

c) An expression accomplishes something, a statement evaluates to a value.

d) Statements are valid in Python, expressions are not.

e) Expressions are valid in Python, statements are not.

1. What is the output of the following code?

my\_str = "I love Python! It is so much fun!"  
if len(my\_str) < 20:  
 print("Python is awesome!")  
else:  
 print("Python is amazing!")

a) *Python is amazing!*

b) Python is awesome!

c) a syntax error

d) nothing will be printed

1. What is the output of the following code when the user inputs “30” at the prompt?

my\_int = int(input())  
if my\_int \* 2 < 10:  
 print("Less than 10!")  
elif my\_int \* 2 < 20":  
 print("Between 10 and 19!")  
elif my\_int \* 2 < 50:  
 print("Between 20 and 49!")  
else:  
 print("Bigger than 50!")

a) Less than 10!

b) Between 10 and 19!

c) Between 20 and 49!

d) *Bigger than 50!*

1. When I run the code below, I get a syntax error. What’s wrong with my code?

my\_variable = input()  
if int(my\_variable) < 50  
 print("Less than 50!")  
else:  
 print("50 or more!")

*There is a missing colon (:) after the if statement on line 2.*

**3.2 Translation**

1. Worin unterscheiden sich Python-Ausdrücke von Python-Anweisungen?

a) Es gibt keinen Unterschied, da es sich um synonyme Bezeichnungen handelt.

b) *Eine Anweisung ist eine Codezeile, die vom Interpreter ausgeführt werden kann. Dagegen handelt es sich bei einem Ausdruck um einen Ausschnitt aus dem Code, dem der Interpreter einen bestimmten Wert zuordnet.*

c) Ein Ausdruck bewirkt etwas, eine Anweisung hat einen Wert.

d) Anweisungen sind in Python gültig, Ausdrücke nicht.

e) Ausdrücke sind in Python gültig, Anweisungen nicht.

1. Welchen Output erzeugt der folgende Code?

my\_str = "I love Python! It is so much fun!"  
if len(my\_str) < 20:  
 print("Python is awesome!")  
else:  
 print("Python is amazing!")

a) *Python is amazing!*

b) Python is awesome!

c) einen Syntaxfehler

d) Es wird nichts ausgegeben.

1. Welchen Output erzeugt der folgende Code, wenn der Benutzer in der Eingabeaufforderung „30“ eingibt?

my\_int = int(input())  
if my\_int \* 2 < 10:  
 print("Less than 10!")  
elif my\_int \* 2 < 20":  
 print("Between 10 and 19!")  
elif my\_int \* 2 < 50:  
 print("Between 20 and 49!")  
else:  
 print("Bigger than 50!")

a) Less than 10!

b) Between 10 and 19!

c) Between 20 and 49!

d) *Bigger than 50!*

1. Bei der Ausführung des nachstehenden Codes meldet der Interpreter einen Syntaxfehler. Was ist die Ursache dieses Problems?

my\_variable = input()  
if int(my\_variable) < 50  
 print("Less than 50!")  
else:  
 print("50 or more!")

*Es fehlt der Doppelpunkt (:) am Ende der if-Anweisung in Zeile 2.*

**3.3**

1. Which of the following will create a range of numbers that includes only the numbers *21*, *22*, *23*, *24*, *25*, *26*, *27*, *28*, and *29*?

a) range(20, 30)

b) *range(21, 30)*

c) range(21, 29)

d) range(20, 29)

e) range("Twenties", "Please")

1. I want to loop through all even numbers from *2* to *20*. Which of the following will do that?

a) for x in range(2, 20):

b) for x in range(2, 21):

c) for x in range(2, 20, 2):

d) *for x in range(2, 21, 2):*

e) for x in range(2, 20, "Even"):

1. I have a variable called “index” and I’ve set its value to 1. If I increment its value by 1 every time through a while loop, what is the proper syntax for a while loop that will run until x > 10?

a) while x > 10:

b) while x >= 10:

c) while x < 10:

d) *while x <= 10:*

e) while x in range(10):

1. What is the difference between break and continue?

a) Using break will interrupt one execution of the loop and start again at the top of the loop, while using continue will exit the loop entirely.

b) They are very similar, but break is used in for loops and continue is used in while loops.

c) They are very similar, but continue is used in for loops and break is used in while loops.

d) *Using continue will interrupt one execution of the loop and start again at the top of the loop, while using break will exit the loop entirely.*

**3.3 Translation**

1. Welche der folgenden Zeilen erzeugt eine Zahlenreihe, die ausschließlich die Elemente *21*, *22*, *23*, *24*, *25*, *26*, *27*, *28* und *29* umfasst?

a) range(20, 30)

*b) range(21, 30)*

c) range(21, 29)

d) range(20, 29)

e) range("Twenties", "Please")

1. Sie möchten eine Schleife erstellen, deren Laufvariable alle geraden Zahlen von *2* bis (einschließlich) *20* durchläuft. Mit welcher der folgenden Zeilen können Sie dies bewerkstelligen?

a) for x in range(2, 20):

b) for x in range(2, 21):

c) for x in range(2, 20, 2):

*d) for x in range(2, 21, 2):*

e) for x in range(2, 20, "Even"):

1. Sie haben eine Variable mit der Bezeichnung x erstellt und ihr den Wert 1 zugewiesen. Nun möchten Sie x zur Programmierung einer while-Schleife nutzen, die den Wert von x bei jedem Durchlauf um 1 anhebt und so lange wiederholt wird, bis x > 10 ist. Wie muss die Kopfzeile dieser Schleife lauten?

a) while x > 10:

b) while x >= 10:

c) while x < 10:

*d) while x <= 10:*

e) while x in range(10):

1. Worin besteht der Unterschied zwischen den Schlüsselwörtern *break* und *continue*?

a) Der break-Befehl unterbricht den aktuellen Durchlauf der Schleife und veranlasst den Interpreter, wieder zum Kopf der Schleife zu springen. Im Gegensatz dazu bewirkt das Schlüsselwort *continue*, dass die Ausführung der Schleife insgesamt abgebrochen wird.

b) Beide Befehle haben eine ähnliche Funktion. Allerdings wird *break* innerhalb von Schleifen eingesetzt, *continue* dagegen außerhalb.

c) Beide Befehle haben eine ähnliche Funktion. Allerdings wird *continue* innerhalb von Schleifen eingesetzt, *break* dagegen außerhalb.

d) *Der continue-Befehl unterbricht den aktuellen Durchlauf der Schleife und veranlasst den Interpreter, wieder zum Kopf der Schleife zu springen. Im Gegensatz dazu bewirkt das Schlüsselwort break, dass die Ausführung der Schleife insgesamt abgebrochen wird.*

**3.4**

1. What is a StopIteration exception in Python?

a) a list of values, sorted in order

b) an object used to loop through values in lists, tuples, or dictionaries

c) *an error code received from the interpreter when moving past the end of a list*

d) a way to build lists from existing lists

1. I have a list called “my\_numbers” that contains the values 4, 8, 15, 16, 23, and 42. I want to loop through that list using an iterator. Each time through the list, I want the variable “next\_value” to contain the next value in the list. Which of the following lines of code will do that?

a) print(my\_numbers[0])

b) for next\_value in range(1, 7):

c) *for next\_value in my\_numbers:*

d) for my\_numbers in next\_value:

1. I have a list called “my\_list” that contains four values (1, 2, 3, and 4). I created an iterator for my\_list called “my\_iterator.” What happens when I call next(my\_iterator) five times in a row?

a) The values 1, 2, 3, 4, and 5 will be printed.

b) The values 1, 2, 3, 4, and 1 will be printed.

c) The values 1, 2, 3, 4 and <nothing> will be printed.

d) *The values* 1*,* 2*,* 3*, and* 4 *will be printed, and then a StopIterator error will be shown.*

1. I have a list called “my\_list” that contains four values (1, 2, 3, and 4). I want to create a new list called “my\_new\_list” from my existing list “my\_list.” I want my\_new\_list to contain only the even values from my\_list, and I want each of those values to be multiplied by 100. Which of the following will accomplish that using a list comprehension?

a) my\_new\_list = [n\*100 for n in my\_list if n.even = True]

b) my\_new\_list = [n for n in my\_list if n% 2 == 0]

c) my\_new\_list = [n for n in my\_list if n.even == True]

d) *my\_new\_list = [n\*100 for n in my\_list if n% 2 == 0]*

**3.4 Translation**

1. Was bezeichnet der Begriff *StopIteration* in Python?

a) eine Liste mit Werten, die nach einer bestimmten Ordnung sortiert sind

b) ein Objekt, das es Ihnen ermöglicht, die Einträge einer Liste, eines Tupels oder eines Dictionary nacheinander aufzurufen

c) *eine Fehlermeldung, die vom Interpreter ausgegeben wird, wenn die Zahl der Iterationsschritte die Zahl der Elemente des Iterators bzw. der diesem zugrunde liegenden Liste übersteigt*

d) eine Methode, mit der sich neue Listen aus bestehenden Listen erstellen lassen

1. Sie haben eine Liste erstellt, die die Bezeichnung *my\_numbers* trägt und die Werte 4, 8, 15, 16, 23 und 42 enthält. Nun möchten Sie diese Liste als Iterator nutzen und der Variablen *next\_value* sukzessive die einzelnen Listenwerte zuweisen. Mit welcher der folgenden Zeilen können Sie dies bewerkstelligen?

a) print(my\_numbers[0])

b) for next\_value in range(1, 7):

***c)*** ***for next\_value in my\_numbers:***

d) for my\_numbers in next\_value:

1. Sie haben eine Liste namens *my\_list* mit insgesamt vier Werten (1, 2, 3 und 4) erstellt. Anschließend haben Sie für diese Liste einen Iterator erzeugt und diesen in der Variablen *my\_iterator* gespeichert. Was passiert, wenn Sie den Befehl next(my\_iterator) fünfmal nacheinander ausführen?

a) Es werden die Werte 1, 2, 3, 4 und 5 ausgegeben.

b) Es werden die Werte 1, 2, 3, 4 und 1 ausgegeben.

c) Es werden die Werte 1, 2, 3, 4 und <nothing> ausgegeben.

d) *Es werden zunächst die Werte* 1*,* 2*,* 3*, and* 4 *ausgegeben. Dann erscheint die Fehlermeldung StopIterator.*

1. In Ihrem Programm gibt es eine Liste namens *my\_list*, die insgesamt vier Werte (1, 2, 3 und 4) umfasst. Nun möchte Sie darauf aufbauend eine neue Liste mit der Bezeichnung *my\_new\_list* erstellen. Dabei soll die Liste *my\_new\_list* nur die geradzahligen Werte aus *my\_list* enthalten, und zwar jeweils mit dem Faktor 100 multipliziert. Bei welcher der folgenden Codezeilen handelt es sich um eine für diesen Zweck geeignete List Comprehension?

a) my\_new\_list = [n\*100 for n in my\_list if n.even = True]

b) my\_new\_list = [n for n in my\_list if n% 2 == 0]

c) my\_new\_list = [n for n in my\_list if n.even == True]

***d)*** ***my\_new\_list = [n\*100 for n in my\_list if n% 2 == 0]***

**4.1**

1. Why do we use functions? (check all that apply)

a) to increase difficulty in programming so fewer people can do it

*b) to eliminate the need to retype code multiple times*

*c) to reduce the potential for bugs by placing repetitive code in one location*

*d) to improve code readability*

e) to get your code to work properly (Python code does not run without functions)

2. I want to define a function called “make\_it\_happen\_captain.” Which of the following will accomplish that task?

a) def make\_it\_happen\_captain:

b) for make\_it\_happen\_captain:

c) while make\_it\_happen\_captain:

d) function make\_it\_happen\_captain:

3. What is the output of the following code?

Def print\_it():  
 print("it")

print("I")  
print("like")  
print\_it()  
print("a lot!")

a) I like it a lot!

b) I like a lot! it

*c) I like it a lot!*

d) I like a lot! it

**4.1 Translation**

1. Warum nutzen Programmierer:innen Funktionen? (Wählen Sie alle zutreffenden Antworten)

a) um das Programmieren schwieriger zu gestalten und dadurch Neulinge abzuschrecken

b) *um die wiederholte Eingabe bestimmter Codeblöcke zu vermeiden*

*c) um die Wahrscheinlichkeit von Bugs zu reduzieren, indem sie vielfach genutzten Code an zentraler Stelle platzieren*

d) *um die Lesbarkeit des Codes zu verbessern*

e) um die Funktionstüchtigkeit des Programms sicherzustellen (Python-Code ist nur dann ausführbar, wenn er Funktionen enthält)

2. Sie möchten eine Funktion mit der Bezeichnung *make\_it\_happen\_captain* definieren. Mit welcher der folgenden Codezeilen können Sie dies bewerkstelligen?

***a)*** ***def make\_it\_happen\_captain:***

b) for make\_it\_happen\_captain:

c) while make\_it\_happen\_captain:

d) function make\_it\_happen\_captain:

3. Welchen Output erzeugt der folgende Code?

Def print\_it():  
 print("it")

print("I")  
print("like")  
print\_it()  
print("a lot!")

a) I like it a lot!

b) I like a lot! it

*c) I like it a lot!*

d) I like a lot! it

**4.2**

1. What is scope?

a) Scope is the range of values a variable can have. An integer that can store values from −100 to 100 is said to have a scope of 200.

b) Scope is the number of variables defined in an application. An application with 8 variables is said to have a scope of 8.

c) *Scope is the sections within code for which a variable or function is defined.*

d) Scope is the type of variables that you can create. In integer scope areas, you cannot create string variables. In string scope areas, you cannot create integers.

1. What is the result of the following code?

a = 1  
b = 2  
c = a + b  
d = a + e  
print(d)

a) 3

b) 4

c) <undefined>

d) *an error*

1. What is the result of the following code?

def my\_function():  
 a = 5  
 b = 5

a = 1  
b = 1  
my\_function()  
print(a + b)

a) *2*

b) 10

c) <undefined>

d) an error

1. What is the result of the following code?

a = 1  
b = 1  
add\_them()  
print(a + b)

def add\_them():  
 a = 2  
 b = 2

a) 2

b) 4

c) <undefined>

d) *an error*

**4.2 Translation**

1. Was ist unter dem Sichtbarkeitsbereich zu verstehen?

a) Der Begriff Sichtbarkeitsbereich bezeichnet den zulässigen Wertebereich einer Variablen. Wenn eine Variable beispielsweise Ganzzahlen aus dem Intervall von -100 bis 100 speichern kann, hat sie einen Sichtbarkeitsbereich von 200.

b) Der Sichtbarkeitsbereich bezeichnet die Zahl der Variablen, die in einer Anwendung definiert werden. Wenn also in einer Anwendung acht Variablen definiert werden, erreicht ihr Sichtbarkeitsbereich einen Wert von 8.

*c)* *Der Sichtbarkeitsbereich ist der Ausschnitt aus dem Programmcode, für den eine Variable oder Funktion definiert ist.*

d) Der Sichtbarkeitsbereich legt fest, welche Typen von Variablen in welchem Segment eines Programms erstellt werden können. Beispielsweise können im integer-Sichtbarkeitsbereich keine String-Variablen erstellt und umgekehrt im String-Sichtbarkeitsbereich keine Variablen mit ganzzahligen Werten definiert werden.

1. Welchen Output erzeugt der folgende Code?

a = 1  
b = 2  
c = a + b  
d = a + e  
print(d)

a) 3

b) 4

c) <undefined>

*d) eine Fehlermeldung*

1. Welchen Output erzeugt der folgende Code?

def my\_function():  
 a = 5  
 b = 5

a = 1  
b = 1  
my\_function()  
print(a + b)

***a) 2***

b) 10

c) <undefined>

d) eine Fehlermeldung

1. Welchen Output erzeugt der folgende Code?

a = 1  
b = 1  
add\_them()  
print(a + b)

def add\_them():  
 a = 2  
 b = 2

a) 2

b) 4

c) <undefined>

*d) eine Fehlermeldung*

**4.3**

1. I want to define a function called “two\_step” that takes two arguments called arg1 and arg2. Which of the following will do that?

a) def two\_step():

b) def two\_step(int arg1, int arg2):

c) two\_step(arg1, arg2):

d) *def two\_step(arg1, arg2):*

1. I want to define a function called “two\_step” that takes two arguments called arg1 and arg2. I want arg2 to have a default value of 12. Which of the following will do that?

a) def two\_step(12):

b) def two\_step(int arg1, int arg2 = 12):

c) def two\_step(arg1, arg2(12)):

d) *def two\_step(arg1, arg2 = 12):*

3. I have a function called “four\_names” that takes four arguments a, b, c, and d. All four arguments are default arguments. Which of the following are valid function calls to four\_names?

a) *four\_names()*

b) *four\_names("Bob")*

c) four\_names("Bob", "Billy")

d) *four\_names(c="Billy")*

4. I want my function to return the value stored in a variable called “result.” I use the line of code return result to do so. What will happen in my code? (Select all correct choices):

a) *The value stored in result will be returned to the point at which the function was called.*

b) The value stored in result will be returned to the top of the function and execution will begin again at that point.

c) After the value is returned, any code in the function block after the return call will be executed.

d) *After the value is returned, any code in the function block after the return call will not be executed.*

**4.3 Translation**

1. Sie möchten eine Funktion mit der Bezeichnung two\_step und zwei Argumenten (arg1 und arg2) definieren. Mit welcher der folgenden Zeilen können Sie dies bewerkstelligen?

a) def two\_step():

b) def two\_step(int arg1, int arg2):

c) two\_step(arg1, arg2):

***d) def two\_step(arg1, arg2):***

1. Sie möchten eine Funktion mit der Bezeichnung two\_step und zwei Argumenten (arg1 und arg2) definieren. Dabei soll arg2 einen Standardwert von 12 erhalten. Mit welcher der folgenden Zeilen können Sie dies bewerkstelligen?

a) def two\_step(12):

b) def two\_step(int arg1, int arg2 = 12):

c) def two\_step(arg1, arg2(12)):

***d) def two\_step(arg1, arg2 = 12):***

3. Sie haben eine Funktion definiert, die die Bezeichnung four\_names trägt und die Eingabe von vier Argumenten (a, b, c und d) fordert. Dabei wurden für alle vier Argumente Standardwerte festgelegt. Welche der folgenden Codezeilen müssen vor diesem Hintergrund als gültige Aufrufe von four\_names gelten?

***a) four\_names()***

***b) four\_names("Bob")***

***c) four\_names("Bob", "Billy")***

***d) four\_names(c="Billy")***

4. Sie möchten dafür sorgen, dass Ihre Funktion den in der Variablen result gespeicherten Wert ausgibt, und nutzen zu diesem Zweck die Zeile return result. Was passiert daraufhin bei der Ausführung Ihres Codes? Wählen Sie alle zutreffenden Antworten:

a) *Der in result gespeicherte Wert wird an der Stelle ausgegeben, an der der Funktionsaufruf erfolgt ist.*

b) Der in result gespeicherte Wert wird an den Kopf der Funktion zurückgemeldet, woraufhin der Interpreter dort mit der Ausführung des Programms fortfährt.

c) Nach der Rückgabe des Werts wird der gesamte Code ausgeführt, der im Funktionskörper auf den Aufruf von return folgt.

d) *Nach der Rückgabe des Werts werden sämtliche Zeilen, die im Funktionskörper auf den Aufruf von return folgen, nicht mehr ausgeführt.*

**5.1**

What are two classifications of errors in Python?

a) *syntax errors and exceptions*

b) syntax errors and compilation errors

c) compilation errors and exceptions

d) programming errors and computer errors

2. What does an “invalid syntax” error message mean?

a) The interpreter executed the code and it came back with an unpredictable result.

b) Something in your program will crash once the application is executed.

c) *Something in your code does not conform to the Python programing syntax rules and cannot be interpreted properly.*

d) Python needs a data type for variables and one was not supplied.

3. What are some ways the interpreter helps you find the cause of a syntax error? (Select all that apply)

a) The interpreter provides an arrow that always points at the exact location of the error.

b) *The interpreter provides the line number of what it believes to be the cause of the error.*

c) The interpreter provides alternative code that would fix the error.

d) *The phrase “syntax error” itself helps you understand that something in the code goes against Python programming rules.*

**5.1 Translation**

Welche zwei Arten von Fehlern müssen in Python unterschieden werden?

a) *Syntaxfehler und Ausnahmen*

b) Syntaxfehler und Kompilierungsfehler

c) Kompilierungsfehler und Ausnahmen

d) Programmfehler und Computerfehler

2. Was bedeutet die Fehlermeldung „invalid syntax“?

a) Bei der Ausführung des Codes durch den Interpreter wurde ein unerwartetes Ergebnis ausgegeben.

b) Ihr Programm wird bei der Ausführung abstürzen.

c) *Mindestens ein Abschnitt in Ihrem Code ist nicht mit der Python-Syntax konform und kann daher nicht durch den Interpreter verarbeitet werden.*

d) Der Interpreter moniert, dass Sie für eine im Programm verwendete Variable keinen Datentyp festgelegt haben, obwohl dies in Python erforderlich ist.

3. Wie hilft Ihnen der Interpreter bei der Identifizierung der Ursache eines Syntaxfehlers? (Wählen Sie alle zutreffenden Antworten)

a) Der Interpreter markiert stets die exakte Position des Fehlers im Code mit einem Pfeil.

b) *Der Interpreter gibt die Nummer der Programmzeile an, in der die Ursache wahrscheinlich zu finden ist.*

c) Der Interpreter unterbreitet Ihnen Vorschläge zur Überarbeitung des Codes.

d) *Durch die angezeigte Meldung „syntax error“ werden Sie darauf hingewiesen, dass Ihr Code die Programmierregeln von Python verletzt.*

**5.2**

1. Which of the following will cause an exception in Python programming?

a) bad syntax in your code

b) *an error that terminates the program during program execution*

c) a variable name that does not conform to Python naming standards or conventions

d) forgetting to put the colon at the end of an if statement

1. Dividing by zero will result in what type of error?

a) SyntaxError

b) *ZeroDivisionError*

c) MathExceptionError

d) IndivisibleError

1. I’m writing code where I’ll be accessing a file. It’s a likely place for a file input/output error to occur. What’s the best way to handle that?

a) You do not need to handle file errors because they are handled by the operating system. Just write your code and do not worry about it.

b) Place all file I/O code in a try/except block and do not specify an error in the except block. This will catch all errors, file-related or otherwise.

c) Place the file open command in the try block of a try/except/finally block. Place the rest of the file-related code in the finally block because once the file is opened, there will be no more errors.

d) *Place all file I/O code in a try/except block and specify file-related errors in the except block. This will catch file errors we anticipate, but nothing else.*

1. What does the finally clause do within a try block?

a) It contains code that executes before a try block to set up error checking and verification on critical variables and functions.

b) It contains code that executes after a try block is interrupted with an exception. If no exception occurs, the code in the finally block does not get executed.

c) It contains code that executes after a try block only if no exception occurs. If an exception occurs within the try block, the code in the finally block does not get executed.

d) *It contains code that executes after a try block regardless of whether an exception occurs. The code in a finally block will always be executed after the try/except block.*

**5.2 Translation**

1. Was löst in Python eine Ausnahme aus?

a) Defizite in Bezug auf die Syntax Ihres Codes

b) *ein Fehler, der die Ausführung des Programms unterbricht*

c) ein Variablenname, der die Python-Standards und Konventionen für die Vergabe von Bezeichnungen verletzt

d) ein fehlender Doppelpunkt am Ende der Kopfzeile einer if-Anweisung

1. Welche Fehlermeldung wird bei einer Division durch null ausgegeben?

a) SyntaxError

*b) ZeroDivisionError*

c) MathExceptionError

d) IndivisibleError

1. Sie schreiben an einem Programm, das auf eine Datei zugreift und dabei eventuell einen Dateieingabe- oder -ausgabefehler auslöst. Wie sollten Sie in diesem Fall am besten vorgehen, um Störungen im Programmablauf zu vermeiden?

a) Sie brauchen sich nicht um dateibezogene Fehler zu kümmern, da diese vom Betriebssystem aufgefangen werden. Deshalb müssen Sie derartige Probleme bei der Arbeit an Ihrem Programm nicht berücksichtigen.

b) Sie platzieren alle Codezeilen mit Dateieingaben und -ausgaben in einer try-Anweisung, deren except-Block keine spezifische Ausnahme benennt. Dadurch fangen Sie alle eventuell auftretenden Fehler auf, ob nun dateibezogen oder nicht.

c) Sie erstellen eine try-Anweisung mit einem try-, einem except- und einem finally-Block. Dann platzieren Sie den open-Befehl im try-Block, die restlichen dateibezogenen Befehle dagegen im finally-Block. So tragen Sie der Tatsache Rechnung, dass nach dem erfolgreichen Öffnen der Datei keine Fehler mehr auftreten können.

d) *Sie platzieren alle Codezeilen mit Dateieingaben und -ausgaben in einer try-Anweisung, deren except-Block spezifische Ausnahmen benennt. Dadurch fangen sie nur die erwarteten Fehler auf.*

1. Was ist der Zweck der finally-Klausel einer try-Anweisung?

a) Sie enthält Codezeilen, die vor dem try-Block ausgeführt werden und der Fehlerdiagnose sowie der Überprüfung kritischer Variablen und Funktionen dienen.

b) Sie enthält Codezeilen, die nur dann ausgeführt werden, wenn der try-Block wegen einer Ausnahme unterbrochen wird. Falls keine Ausnahme auftritt, wird der Code in der finally-Klausel nicht ausgeführt.

c) Sie enthält Codezeilen, die nur dann ausgeführt werden, wenn im try-Block keine Ausnahme auftritt. Falls eine Ausnahme ausgelöst wird, wird der Code in der finally-Klausel nicht ausgeführt.

d) *Sie enthält Codezeilen, die in jedem Fall im Anschluss an den try- bzw. except-Block ausgeführt werden – unabhängig davon, ob eine Ausnahme auftritt oder nicht.*

**5.3**

1. There are five levels of logging criticality. What are those levels in order from least to most critical?

a) *debug, info, warning, error, critical*

b) critical, error, warning, info, debug

c) info, debug, warning, error, critical

d) critical, error, warning, debug, info

1. If we want to use logging, we must use the statement: import logging. What does that statement do?

a) signals to the interpreter that you want to use logging and it should turn on the logging functionality

b) *imports logging libraries that you can use in Python code*

c) creates a generic set of log files that you can read from and write to

d) imports a set of logging code that will automatically create log messages at the beginning and ending of each function

1. Which line of code below will set the log level to Warning?

a) logging.level = logging.WARNING

b) logging.basicConfig.level = logging.WARNING

c) logging.basicConfig = logging.WARNING

d) *logging.basicConfig(level = logging.WARNING)*

4. Besides using whatever text you wish to describe the error, is there a way to further customize the log messages?

a. No, you can only customize by setting the text message.

b. *Yes, you can pass in attributes representing things such as time of the error, line number, process ID, etc.*

c. Yes, you can change the logging method to log through the interpreter’s subprocesses instead of the main process.

d. Yes, you can turn on automatic logging, which enables custom messages without having to write any code.

**5.3 Translation**

1. Im Logging-Modul von Python werden fünf Dringlichkeitsstufen unterschieden. Wie lauten diese in der richtigen Reihenfolge, beginnend mit der niedrigsten?

***a) debug, info, warning, error, critical***

b) critical, error, warning, info, debug

c) info, debug, warning, error, critical

d) critical, error, warning, debug, info

1. Damit Sie die Protokollierungsfunktionen in Ihrem Programm nutzen können, muss dieses die folgende Zeile enthalten: import logging. Was bewirken Sie mit dieser Anweisung?

a) Sie signalisieren dem Interpreter, dass Sie den Programmablauf protokollieren möchten und er die dafür nötigen Funktionen aktivieren soll.

b) *Sie importieren die logging-Bibliothek, deren Befehle dann zur Verwendung in Ihrem Python-Code zur Verfügung stehen.*

c) Sie erstellen ein Set generischer Logdateien, die Sie auslesen und beschreiben können.

d) Sie importieren vordefinierte Codezeilen, die automatisch am Anfang und Ende jeder Funktion eingefügt werden und jeweils passende Protokolleinträge erzeugen.

1. Mit welcher Codezeile können Sie die Dringlichkeitsschwelle auf WARNING setzen?

a) logging.level = logging.WARNING

b) logging.basicConfig.level = logging.WARNING

c) logging.basicConfig = logging.WARNING

***d) logging.basicConfig(level = logging.WARNING)***

4. Gibt es – abgesehen von den in den logging-Zeilen enthaltenen Ereignisbeschreibungen – noch andere Möglichkeiten zur individuellen Anpassung der Protokolleinträge?

a. Nein, die Gestaltung der Textnachrichten zur Beschreibung der erfassten Ereignisse ist die einzige Möglichkeit zur individuellen Anpassung der Protokolleinträge.

b. *Ja, Sie können mithilfe von Attributen veranlassen, dass zusätzlich Zeit- und Datumsangaben, die betreffende Zeilennummer, die Prozesskennung und andere Daten erfasst werden.*

c. Ja, Sie können den Interpreter anweisen, die Protokollierung nicht über den Hauptprozess, sondern über die Unterprozesse laufen zu lassen.

d. Ja, Sie können die automatische Protokollierung aktivieren. Dabei werden individuell angepasste Meldungen ausgegeben, ohne dass Sie auch nur eine Codezeile schreiben müssen.

**6.1**

1. Which of the following are likely reasons to create your own module in Python? (Select all that apply)

a) You want your code to run faster.

b) *You want to be able to reuse code in a different application.*

c) *You want to package a library of code for others to use.*

d) You want to reduce syntax errors and exceptions.

2. True or false? Modules allow you to take code and reuse it in multiple applications.

a) *true*

b) false

**6.1 Translation**

1. Welche Vorteile hat die Erstellung eigener Python-Module? (Wählen Sie alle zutreffenden Antworten)

a) Die Ausführungsgeschwindigkeit Ihres Codes wird verbessert.

b) *Der im Modul enthaltene Code kann für verschiedenen Anwendungen bereitgestellt werden.*

c) *Der im Modul enthaltende Code kann als Bibliothek anderen Programmierer:innen zugänglich gemacht werden.*

d) Die Zahl der Syntaxfehler und Ausnahmen wird reduziert.

2. Wahr oder falsch: Module eröffnen Ihnen die Möglichkeit, Code für verschiedene Anwendungen bereitzustellen.

*a) wahr*

b) falsch

**6.2**

1. What is the purpose of a namespace?

a) enables faster applications by reducing interpreter code-analysis time

b) *provides a way to make variable and function names unique and prevent their duplication across modules and packages*

c) reduces the likelihood of an exception by automatically handling exceptions at different levels

d) removes the need to import libraries

1. Morgan and Kyle developed a soccer statistics library for Python called “soccerstats.” We want to import a function from that library called “score\_chance.” Which line of code will import only that function?

a) import soccerstats

b) import score\_chance

c) import score\_chance from soccerstats

d) *from soccerstats import score\_chance*

1. What if we wanted to import the entire soccerstats library? Which line of code will import that library?

a) *import soccerstats*

b) import score\_chance

c) import all from soccerstats

d) from soccerstats import all

1. What is the benefit of importing an entire library rather than just importing selected functions from that library?

a) *Importing the entire library reduces the likelihood that a name conflict will exist in the global namespace by making use of the dot notation.*

b) Importing the entire library reduces the likelihood that a name conflict will exist in the local namespace.

c) Importing the entire library reduces the likelihood that a name conflict will exist in the built-in namespace.

d) There is no benefit.

**6.2 Translation**

1. Welchem Zweck dienen Namensräume?

a) Namensräume beschleunigen die Ausführung von Python-Programmen, weil sie die vom Interpreter zur Analyse des Codes benötigte Zeit reduzieren.

b) *Namensräume erleichtern die Vergabe eindeutiger Variablen- und Funktionsnamen und helfen bei der Vermeidung von Doppelungen – besonders beim Import zusätzliche Module und Pakete.*

c) Namensräume senken das Risiko eines durch Ausnahmen bedingten Programmabbruchs, weil sie automatische Mechanismen zur Ausnahmebehandlung auf verschiedenen Programmebenen beinhalten.

d) Namensräume machen die Einbindung zusätzlicher Bibliotheken überflüssig.

1. Lea und Mehmet haben eine Bibliothek namens soccerstats entwickelt, die Funktionen für statistische Fußballanalysen enthält. Welche der folgenden Codezeilen muss ein Programm enthalten, das ausschließlich die Funktion score\_chance aus der fraglichen Bibliothek importiert?

a) import soccerstats

b) import score\_chance

c) import score\_chance from soccerstats

***d) from soccerstats import score\_chance***

1. Wie müsste die Codezeile lauten, wenn stattdessen die komplette Bibliothek importiert werden soll?

***a) import soccerstats***

b) import score\_chance

c) import all from soccerstats

d) from soccerstats import all

1. Welchen Vorteil hat der Import einer kompletten Bibliothek gegenüber dem Import ausgewählter Funktionen aus dieser Bibliothek?

a) *Die Einbindung der kompletten Bibliothek senkt das Risiko eines Namenskonflikts mit dem globalen Namensraum, da die importierten Funktionen hier nur in punktierter Schreibweise aufgerufen werden können.*

b) Die Einbindung der kompletten Bibliothek senkt das Risiko eines Namenskonflikts mit dem lokalen Namensraum.

c) Die Einbindung der kompletten Bibliothek senkt das Risiko eines Namenskonflikts mit dem Built-in-Namensraum.

d) Es gibt keine nennenswerten Vorteile.

**6.3**

1. Which of the following is a valid comment in Python?

a) Comment: This is a comment!

b) // This is a comment!

c) *# This is a comment!*

d) -- This is a comment!

1. Which of the following is a single-line docstring in Python?

a) Docstring: This is a docstring!

b) /\* This is a docstring! \*/

c) # This is a docstring! #

d) *""" This is a docstring! """*

1. I have documented a function called “foo” with a docstring. Which of the following will display my function and associated docstring?

a) foo()

b) # foo()

c) print(foo.docstring)

d) *help(foo)*

**6.3 Translation**

1. Bei welcher der folgenden Codezeilen handelt es sich um einen gültigen Kommentar in Python?

a) Comment: This is a comment!

b) // This is a comment!

***c) # This is a comment!***

d) -- This is a comment!

1. Bei welcher der folgenden Codezeilen handelt es sich um einen einzeiligen Docstring in Python?

a) Docstring: This is a docstring!

b) /\* This is a docstring! \*/

c) # This is a docstring! #

***d) """ This is a docstring! """***

1. Sie einen Docstring in eine Funktion namens *foo* eingefügt. Mit welcher der folgenden Zeilen können Sie die Syntax der Funktion und den dazugehörigen Docstring anzeigen?

a) foo()

b) # foo()

c) print(foo.docstring)

***d) help(foo)***

**6.4**

1. Which library provides a wide range of graphing functions to visualize data?

a) NumPy

b) *Matplotlib*

c) SciPy

d) pandas

e) scikit-learn

1. Which library provides machine-learning algorithms for Python programmers?

a) NumPy

b) Matplotlib

c) SciPy

d) pandas

e) *scikit-learn*

1. Which library provides the fast, efficient, and easy-to-use DataFrame data structure?

a) NumPy

b) Matplotlib

c) SciPy

d) *pandas*

e) scikit-learn

1. Which library provides a wide range of data types and functions for programming data science applications in Python?

a) *NumPy*

b) Matplotlib

c) SciPy

d) pandas

e) scikit-learn

1. Which library provides scientific functions for linear algebraic functions, interpolation, signal processing, and more?

a) NumPy

b) Matplotlib

c) *SciPy*

d) pandas

e) scikit-learn

**6.4 Translation**

1. Welche der folgenden Bibliotheken stellt umfangreiche Funktionen zur Erstellung von Graphen und anderweitigen Visualisierung von Daten bereit?

a) NumPy

***b) Matplotlib***

c) SciPy

d) Pandas

e) Scikit-Learn

1. Welche der folgenden Bibliotheken bietet Python-Programmierer:innen Algorithmen für das maschinelle Lernen?

a) NumPy

b) Matplotlib

c) SciPy

d) Pandas

***e) Scikit-Learn***

1. Welche Bibliothek beinhaltet die schnelle, effiziente und nutzerfreundliche Datenstruktur DataFrame?

a) NumPy

b) Matplotlib

c) SciPy

***d) Pandas***

e) Scikit-Learn

1. Welche der folgenden Bibliotheken umfasst vielfältige Datentypen und Funktionen zur Programmierung datenwissenschaftlicher Anwendungen in Python?

***a) NumPy***

b) Matplotlib

c) SciPy

d) Pandas

e) Scikit-Learn

1. Welche der folgenden Bibliotheken bietet Ihnen Funktionen für lineare Algebra, Interpolation, die Signalverarbeitung und viele andere Anwendungsbereiche?

a) NumPy

b) Matplotlib

***c) SciPy***

d) Pandas

e) Scikit-Learn