**Part 1: Matching – Each answer is one-to-one**

*Matching 1*

*Draw a line from the spaghetti code function to the optimized function*

|  |  |  |
| --- | --- | --- |
| Q11. |  |  |
| Q12. |  |  |
| Q13. |  |  |
| Q14. |  |  |

*Matching 2*

*Draw a line from the data size to the approximate byte conversion (1e3 == 103)*

|  |  |  |
| --- | --- | --- |
| Q1. | 128 KB | ~1.28e15 bytes |
| Q2. | 128 PB | ~1.28e9 bytes |
| Q3. | 128 GB | ~1.28e12 bytes |
| Q4. | 128 TB | ~1.28e3 bytes |
| Q5. | 128 MB | ~1.28e6 bytes |

*Matching 3*

*Given the list shown, draw a line from the code to the sorted output*



|  |  |  |
| --- | --- | --- |
| Q6. |  |  |
| Q7. |  |  |
| Q8. |  |  |
| Q9. |  |  |
| Q10. |  |  |

*Matching 4*

*Using the file directory below, draw a line from the code snippet to the returned output*

*Note: The following code is run in the file labeled “f4\_code.py”*

A computer code with blue text

Description automatically generated

|  |  |  |
| --- | --- | --- |
| Q15. |  | True |
| Q16. |  | True |
| Q17. |  | True |
| Q18. |  | False |
| Q19. |  | False |
| Q20. |  | False |

*Matching 5*

*For each code snippet, the printed output is shown in a comment to the right of it.*

*Draw a line to the matching function definition to the code called to produce the output*

|  |  |  |
| --- | --- | --- |
| Q21. |  | endswith |
| Q22. |  | get |
| Q23. |  | find |
| Q24. |  | count |
| Q25. |  | index |

**PART 2: Free Response -- Computers**

*Write your answer to each question on the line given below it.*

Q26. Name 4 standard Python modules (i.e. libraries built-in with Python.)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Q27. What are the 3 general types of programming languages?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Q28. What does ALU stand for?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Q29. Name the phenomenon where processor speeds, memory, and overall computer capacity doubles every 1-2 years?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Q30. Name 5 different machines with operating systems in them

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**PART 2: Free Response – Recursion Analysis**

**For each question, what is the printed output?**

*Write your answer on the line to the right of the question*

|  |  |  |
| --- | --- | --- |
| Q31. |  |  |
| Q32. |  |  |
| Q33. |  |  |
| Q34. |  |  |
| Q35. |  |  |
| Q36. |  |  |

**PART 2: Free Response – File I/O**

*Using the 2 files given below, write your answer to the questions on the line to the right.*

*Note: Assume the file contents are reset to the original content after each question*

*Note 2: None of the code in this section will return an error*

|  |  |
| --- | --- |
| ***heroes.json*** | ***allofme.txt*** |
| {  "team": "Sonic Heroes",  "members": [  {  "name": "Sonic",  "alias": "The Hedgehog",  "powers": ["Speed", "Agility", "Spin Dash"]  },  {  "name": "Tails",  "alias": "The Fox",  "powers": ["Genius", "Fly", "Acrobatics"]  },  {  "name": "Knuckles",  "alias": "The Echidna",  "powers": ["Strength", "Glide", "Dig"]  }  ]  } | I see no, hear no evil  Black writings on the wall  Unleash a million faces  And one by one they fall  Black-hearted evil  Brave-hearted hero  I am all, I am all I am |

|  |  |  |
| --- | --- | --- |
| Q37. | What is printed in the following code? |  |
| Q38. | What is printed in the following code? |  |

|  |  |  |
| --- | --- | --- |
| Q39. | What is printed in the following code? |  |
| Q40. | What is the contents of allofme.txt after this code is ran? (write each line separately) | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

|  |  |  |
| --- | --- | --- |
| Q41. | What is printed in the following code? |  |
| Q42. | What is printed in the following code?  (write each line separately) | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

**PART 2: Free Response – OOP Analysis**

*Use the following code snippet to answer questions QX – QX below*

A screen shot of a computer program

Description automatically generated

Q43. List all of the *setter* methods for the object *ticon (*instance of the Pencil class)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Q44. List the names of all of the *attributes* for the object *ticon (*instance of the Pencil class)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Q45. What is printed in the following line of code?



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Q46. What is printed in the following line of code?



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Q47. What is the printed output of this code snippet?

A close up of text

Description automatically generated

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Q48-Q50. Using the lines below, create a new Pencil instance with the brand “Paper Mate.” Write more code such that the following output is produced after calling write() on the instance (DO NOT directly modify the value.)



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Q51-Q55. In the space below, write a new class based on the following requirements:

**MechanicalPencil class:**

- Inherits from Pencil  
- Initialized with a “lead\_amt” parameter (float) that has a default value of 3.0  
- Brand value is “BIC”, thickness value is “0.5mm”  
- Calls the parent class initializer, passing the appropriate parameter values  
- Override the write() function to be the following:

*If num\_lead > 0, 1) call the parent class’s write() method; 2) Subtract 0.1 from num\_lead*

*Otherwise, print “Out of pencil lead…”*

**PART 3: Multiple Choice – Big O / Search / Sort**

*Circle the answer choice for each question given*

Q56. What is the Big-O of the following function?

|  |  |  |  |
| --- | --- | --- | --- |
|  | | |  |
| 1. O(1) | 1. O(n) | 1. O(n2) | 1. O(2n) |

Q57. What is the Big-O of the following function?

|  |  |  |  |
| --- | --- | --- | --- |
|  | | |  |
| 1. O(1) | 1. O(n) | 1. O(n2) | 1. O(2n) |

Q58. What is the Big-O of the following function?

|  |  |  |  |
| --- | --- | --- | --- |
|  | | |  |
| 1. O(1) | 1. O(n) | 1. O(n2) | 1. O(2n) |

Q59. What is the Big-O of the following function?

|  |  |  |  |
| --- | --- | --- | --- |
|  | | |  |
| 1. O(1) | 1. O(n) | 1. O(n2) | 1. O(2n) |

Q60. What is the Big-O of the following function?

|  |  |  |  |
| --- | --- | --- | --- |
|  | | |  |
| 1. O(1) | 1. O(n) | 1. O(n2) | 1. O(nlogn) |

Q61. What is the Big-O of the following function?

|  |  |  |  |
| --- | --- | --- | --- |
|  | | |  |
| 1. O(1) | 1. O(n) | 1. O(n2) | 1. O(2n) |

Q62. What is the Big-O of the following function?

|  |  |  |  |
| --- | --- | --- | --- |
|  | | |  |
| 1. O(1) | 1. O(n) | 1. O(n2) | 1. O(2n) |

Q63. Which sorting algorithm runs in O(nlogn) runtime?

|  |  |  |  |
| --- | --- | --- | --- |
| 1. Bubble | 1. Insertion | 1. Selection | 1. Merge |

Q64. **(Circle all the apply)** Which sorting algorithm runs in O(n2) runtime?

|  |  |  |  |
| --- | --- | --- | --- |
| 1. Selection | 1. Monkey | 1. Bogo | 1. Insertion |

Q65. Which algorithm recursively splits the array into 2 sublists and sorts on the sublists?

|  |  |  |  |
| --- | --- | --- | --- |
| 1. Quick | 1. Merge | 1. Half | 1. Permutation |

Q66. Which sorting algorithm swaps the *i-th* smallest value with the *i-th* value in the list?

|  |  |  |  |
| --- | --- | --- | --- |
| 1. Selection | 1. Bubble | 1. Quick | 1. Insertion |

Q67. What type of sorting algorithm does Python 2.3-3.10 built-in ‘sort’ function use?

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1. Julian |  | 1. Quick | 1. Tim | 1. Snake |

Q68. Which of these search algorithms works only with a sorted numeric array?

|  |  |  |  |
| --- | --- | --- | --- |
| 1. Needle | 1. Binary | 1. Linear | 1. Quick |

Q69. **(Circle all the apply)** Which of these functions are built-in Python search functions?

|  |  |  |  |
| --- | --- | --- | --- |
| 1. find | 1. look | 1. match | 1. index |

**PART 4: Code Writing -- Recursion 1 (Q70-Q75)**

**[5pts 🡪 base case (1); recursive call (1); no stack overflow (1); correct (2)]**

*Using the space below, write a recursive function that converts all characters (excluding spaces) that do not match the very first letter into a period (.)*

*Use the following test cases as reference. You can also use default parameters if you wish.*

A close up of text

Description automatically generated

**PART 4: Code Writing -- Recursion 2 (Q76-Q80)**

**[5pts 🡪 base case (1); recursive call (1); no stack overflow (1); correct (2)]**

*Using the space below, write a recursive function that converts all numbers into the letter n*

*Use the following test cases as reference. You can also use default parameters if you wish.*

A close up of text

Description automatically generated

**PART 4: Code Writing -- Recursion 3 (Q81-Q85)**

**[5pts 🡪 base case (1); recursive call (1); no stack overflow (1); correct (2)]**

*Using the space below, write a recursive function that censors a word – i.e. retains the first letter of a word and turns the rest into asterisks (\*).*

*Use the following test cases as reference. You can also use default parameters if you wish.*

A text on a white background

Description automatically generated

**PART 4: Code Writing – OOP 1 (Q86-Q90)**

**[5pts 🡪 attributes (1), methods (1), instance (1), correct (2)]**

*Using the space below, create a new class definition based on the attributes and methods described. Then create an instance of the class.*

|  |  |
| --- | --- |
| **Toothpaste** | |
| *Attributes (assigned in \_\_init\_\_)* | *Methods* |
| Parameter attributes   * brand : str * flavor : str (default ‘mint’)   Pre-set attributes   * amount : int = 100 | * **Getter methods** for brand and amount * **Setter method** for amount * **Custom method** called *squeeze* that prints “Squeezing [flavor] [brand] toothpaste” if the *amount* is greater than 0 and subtracts 2 from *amount*. |

**Instance**: Create an instance of toothpaste with the brand “Colgate” and flavor “cool mint”. Set the amount to 10 and call squeeze() such that it prints “Squeezing cool mint Colgate toothpaste.”

**PART 4: Code Writing – OOP 2 (Q91-Q95)**

**[5pts 🡪 attributes (1), methods (1), instance (1), correct (2)]**

*Using the space below, create a new class definition based on the attributes and methods described. Then create an instance of the class.*

|  |  |
| --- | --- |
| **Television** | |
| *Attributes (assigned in \_\_init\_\_)* | *Methods* |
| Parameter attributes   * year : int * size : tuple * channels : int (default = 100)   Pre-set attributes   * on : bool = False | * **Getter methods** for year * **Custom method** called *subscribe* that takes in as input a list of channels (strings) and increases the channels value by the number of elements. Print out “Added [name] channel” for each element in the list. * **Custom method** called *power* that toggles the on value (set to True if False, set to False if True) |

**Instance**: Create an instance of Television with the year = 1998, size = (30, 23, 19), and channels = 100. Call subscribe() on the instance with the list parameter value = [“HBO”, “Cartoon Network”] such that it prints out “Added HBO channel” and “Added Cartoon Network channel”

**PART 4: Code Writing – OOP 3 (Q96-Q100)**

**[5pts 🡪 attributes (1), methods (1), instance (1), correct (2)]**

*Using the space below, create a new class definition based on the attributes and methods described. Then create an instance of the class.*

|  |  |
| --- | --- |
| **Dragon** | |
| *Attributes (assigned in \_\_init\_\_)* | *Methods* |
| Parameter attributes   * name : str * colors : list   Pre-set attributes   * hoard = None | * **\_\_str\_\_ method** that returns the name, colors, and hoard information * **Setter method** for hoard * **Custom method** called *fire* that prints that the name of the dragon breathing a randomly chosen color fire. (i.e. “[name] breathed [color] fire”) if there are more than 0 elements in *colors* * **Custom method** called *protect* that prints “[name] protects their hoard of [hoard]” if hoard is not None. Otherwise print “[name] protects nothing”. |

**Instance**: Create an instance of Dragon named “Saphira” whose elements are “blue” ,“green”, “red”. Change the instance’s hoard to “dragon eggs” using the setter method defined. Call the instance’s protect method such that it prints “Saphira protects their hoard of dragon eggs”

**ANSWER KEY**

**PART 1**

MATCHING 1

A screenshot of a computer code

Description automatically generatedA computer screen shot of a code

Description automatically generated

A screen shot of a computer program

Description automatically generatedA screenshot of a computer code

Description automatically generated

MATCHING 2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1. ~1.28e3 b | 2. ~1.28e15 b | 3. ~1.28e9 b | 4. ~1.28e12 b | 5. ~1.28e6 b |

MATCHING 3

A close-up of a name

Description automatically generated

MATCHING 4

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 15. True | 16. False | 17. True | 18. True | 19. False | 20. False |

MATCHING 5

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 21. count | 22. find | 23. endswith | 24. index | 25. get |

**PART 2**

*FREE RESPONSE – Computers*

Q26. See Lecture 13 – Slide 26

Q27. Machine, Assembly, High Level

Q28. Arithmetic and Logic Unit

Q29. Moore’s Law

Q30. See Lecture 13 – Slide 15

*FREE RESPONSE -- Recursion*

A close-up of numbers and symbols

Description automatically generated

*FREE RESPONSE – File I/O*

A close up of a text

Description automatically generated

Q40.

A black background with white text

Description automatically generated

A close up of words

Description automatically generated

*FREE RESPONSE – OOP Analysis*

Q43. set\_thickness, set\_color

Q44. Color, thickness, brand, wrote

Q45. 

Q46. 

Q47. 

Q48 – Q50.

A close up of text

Description automatically generated

Q51-Q55.

A screen shot of a computer code

Description automatically generated

**PART 3**

Multiple Choice

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Q56. B | Q57. A | Q58. C | Q59. A | Q60. C |
| Q61. B | Q62. A | Q63. D | Q64. A,D | Q65. B |
| Q66. A | Q67. C | Q68. B | Q69.A,D |  |

**PART 4**

Recursion 1 (Sample answer)

A screen shot of a computer code

Description automatically generated

Recursion 2 (Sample answer)

A white background with colorful text

Description automatically generated

Recursion 3 (Sample answer)

A screen shot of a computer

Description automatically generated

OOP 1 (Sample answer)

A screen shot of a computer program

Description automatically generated

OOP 2 (Sample answer)

A screen shot of a computer program

Description automatically generated

OOP 3 (Sample answer)

A screenshot of a computer program

Description automatically generated