

## CS 149: Programming Fundamentals James Madison University Written Exam #1

*This work complies with the JMU Honor Code. I have neither given nor received unauthorized assistance, and I will not discuss the exam contents with anyone who has not taken it for credit.*

Name: \_\_\_\_\_ Signature: \_\_\_\_\_

1. (12 points) Indicate whether each of the following statements is true or false:

- i. \_\_\_\_\_ `def`, `return`, and `import` are all examples of Python Language Keywords.
- ii. \_\_\_\_\_ In the Python statement `x = input("Enter number: ")`, `input` is the name of a function.
- iii. \_\_\_\_\_ The scope of a local variable does not extend beyond the function body.
- iv. \_\_\_\_\_ All functions must include a return statement.
- v. \_\_\_\_\_ Comments can be used by the Python interpreter to help find logic errors in a program.
- vi. \_\_\_\_\_ This is a stylistically-correct Python variable name: `My_list`

2. (12 points) Choose the best answer to each of the following:

- i. \_\_\_\_\_ In order to run, a Python application must have
  - a) valid syntax
  - b) docstrings
  - c) inline comments
  - d) integer variables
- ii. \_\_\_\_\_ When calling a function, the arguments are
  - a) specialized variables introduced in the first line of the function definition
  - b) the values passed to the function
  - c) both of the above
  - d) neither of the above
- iii. \_\_\_\_\_ The assignment operator is used to
  - a) indicate the type of a variable
  - b) combine two data values

- c) store an object in a variable
- d) determine the remainder

3. (12 points) Vocabulary Matching

_____ variable	A) The region of the program where a variable is accessible.
_____ expression	B) The order in which certain operations are evaluated.
_____ literal	C) Newlines, tabs, and other “invisible” characters.
_____ operator	D) A segment of code that evaluates to some value.
_____ module	E) A Python file that may be imported into another Python program.
_____ precedence	F) A symbol that represents a computation like addition.
_____ scope	G) An object that is written directly into the code such as the 3 in the statement <code>x = 3</code> .
_____ whitespace	H) A named item used to store a value.

4. (15 points) Write a Python statement that:

- i. Assigns 7 to the variable `count`.

---

- ii. Calls a function named `happy`, passing the argument 3, and assigning the return value to a variable named `sad`.

---

- iii. Creates a dictionary containing the key “Bob” mapped to the value 24 and assigns the result to a variable named `ages`.

---

iv. Adds 54 to the dictionary stored in `ages` with the key "Alice".

---

v. Creates a list containing the strings "red", "green" and "blue" and stores it in a variable named `colors`.

---

vi. Prints the last entry in the list stored in `colors`.

5. (12 points) Evaluate the following expressions, and indicate the data type and value of the result. Write ERROR in both columns if there is a syntax error.

	Type	Value
<code>-5 + 1</code>		
<code>int(-(5 / 4))</code>		
<code>1 + 6 + " is " + 3 + 4</code>		
<code>15 % 4</code>		
<code>"Hello" * 2</code>		
<code>(9 / 2.0) * 2</code>		

6. (15 points) What is the output of the following program? (It compiles and runs without error.)

```
def rectangle_area(width, length):  
    area = width * length  
    return 10  
  
print("R1: ", rectangle_area(300,  
10))  
print("R2: ", rectangle_area(20,
```

```
400))
print("R3: ", rectangle_area(20, 20))
```

Draw a memory diagram to show the contents of memory just before `rectangle_area` returns for the first time

(12 points) Complete the following function. You may assume that no trip lasts for 24 hours or more. You may also assume that the parameters are all correct and reasonable (i.e., that `miles_end` is greater than or equal to `miles_start`).

```
"""Computes a car's average speed over the length of a trip.

    Args:
        miles_start (int): odometer reading at the start
of the trip
        miles_end (int): odometer reading at the end of the trip
        hrs_start (int): hours on the (24 hour) clock at the start
        mins_start (int): minutes on the clock at the start
        hrs_end (int): hours on the (24 hour) clock at the end
        mins_end (int): minutes on the clock at the end

    Returns:
        Float: the average speed (in miles per hour)
"""
def average_Speed(miles_start, miles_end, hrs_start,
mins_start, hrs_end,
mins_end):
    pass
```

(10 points) Complete the following table based on the `average_speed` function above. (You may assume that the method works properly, even if you were unable to answer the previous question.)

<code>miles_start</code>	<code>miles_end</code>	<code>hrs_start</code>	<code>mins_start</code>	<code>hrs_end</code>	<code>mins_end</code>	<code>return</code>
15000	15030	14	15	14	45	
15000	15030	14	45	15		60.0
80100	80175		0	3	0	75.0
80005	80015	1	30	2	0	
60000	60010	7		7	30	40.0