

## CS 141

### MIDTERM 1 PRACTICE PROBLEMS

1. The \_\_\_\_\_ function reads a piece of data that has been entered at the keyboard and returns that piece of data, as a string, back to the program.
  - a. input
  - b. output
  - c. eval\_input
  - d. string\_input
2. In a print statement, you can set the \_\_\_\_\_ argument to a space or empty string to stop the output from advancing to a new line.
  - a. stop
  - b. end
  - c. separator
  - d. newline
3. What type of data is being stored into the variable **sold** in the following line of code? `sold = 256.752`
  - a. int
  - b. float
  - c. str
  - d. boolean
4. What type of data is being stored into the variable **a** in the following line of code? `a = input("Enter a number: ")`
  - a. int
  - b. float
  - c. str
  - d. boolean
5. After the execution of the following statement, the variable **price** will reference the value \_\_\_\_\_. `price = int(68.549)`
  - a. 68
  - b. 69
  - c. 68.55
  - d. 68.54
6. It is recommended that programmers should avoid using \_\_\_\_\_ variables in a program when possible.
  - a. local
  - b. global
  - c. string global
  - d. keyword

Questions 7-10 refer to the following code (line numbers present for reference).

```
1 def average(first, second, third):
2     avg = (first + second + third) / 3
3     print("average is", avg)
4
5 def main():
6     x = float(input("First number? "))
7     y = float(input("Second number? "))
8     z = float(input("Third number? "))
9     average(x, y, z)
10
11 main()
```

7. Line 1 is the function \_\_\_\_\_ for the function average.
- call
  - header
  - block
  - parameter
8. In Lines 2 and 3, avg is a \_\_\_\_\_ variable to the function average.
- global
  - constant
  - defined
  - local
9. In Line 1, first, second and third are \_\_\_\_\_ for the function average.
- headers
  - returns
  - parameters
  - arguments
10. In Line 9, x, y, and z are \_\_\_\_\_ used when calling the average function from main.
- headers
  - returns
  - parameters
  - arguments

11. What is the result of the following Boolean expression, if x equals 5, y equals 3, and z equals 8?  
 $x < y$  or  $z > x$
- a. True
  - b. False
  - c. 8
  - d. 5
12. What is the result of the following Boolean expression, if x equals 5, y equals 3, and z equals 8?  
 $\text{not } (x < y \text{ or } z > x) \text{ and } y < z$
- a. True
  - b. False
  - c. 8
  - d. 5
13. The expression `print(str(8) + str(9))` will output \_\_\_\_\_.
14. The result of the expression `11.3 + 6.6` is \_\_\_\_\_.
15. What is output for the following line of code? \_\_\_\_\_  
`print(format(76.15854, '.3f'))`
16. A(n) \_\_\_\_\_ refers to a sequence of well-defined steps to solve a problem.
17. A(n) \_\_\_\_\_ statement will execute one block of statements if its condition is true, or another block if its condition is false.
18. A(n) \_\_\_\_\_-controlled loop causes a statement or set of statements to repeat as long as a condition is true.
19. \_\_\_\_\_ are notes of explanation that document lines or sections of a program.

20. What is x after the following statements?

```
x = 2
x *= x + 3
```

21. What is the output for y?

```
y = 0
for i in range(2, 9):
    y += i

print(y)
```

22. What will be displayed after the following loop terminates?

```
number = 25
isPrime = True
i = 2
while i < number and isPrime:
    if number % i == 0:
        isPrime = False

    i += 1

print("i is", i, "isPrime is", isPrime)
```

23. The following code displays \_\_\_\_\_.

```
age = 19

if age < 18:
    print("Minor")
elif age >= 18 and age < 65:
    print("Adult")
else:
    print("Senior Citizen")
```

24. Write code that will randomly generate a number between 0 and 100. If that number is greater than 50, output that it is "Too high", otherwise, output "Too low".

25. Write a function called `calculate_total_bill` that takes in two parameters `bill_amt` and `perc_tip` (15% entered as `.15`) and **returns** (not prints) the cost of the bill with tip added.

26. Write a function called `compareNumbers` that takes in 2 parameters `num1` and `num2` and outputs (prints) the numbers in ascending order.

27. Given that  $n$  refers to a positive integer, write a loop to compute the sum of the squares of the first  $n$  counting numbers, and associate this value with `total`. Thus if  $n$  equals 4, your code should put  $1*1 + 2*2 + 3*3 + 4*4$  into `total`.

28. Write a loop that asks the user to enter a series of positive numbers. The user should enter a negative number to signal the end of the series. The program should output whether each number entered is even or odd.