


Question	Out of	Mark
1. True/False	15	15
2. Multiple Choice	15	14
3. Short Answers	30	28
4. Logic Errors	20	16
5. A Little Code	20	14
Total	100	87

Question 1: True/False – 15 Marks (1 each)

For the following questions, please circle (or indicate as specified by the question) your answer directly on the exam sheet. Note that questions are each worth one point unless otherwise indicated.

- 1) A variable in Python has a name and a location (memory). ☒ True ☐ False
 - 2) Python can have variables that are integers. ☒ True ☐ False
 - 3) Boolean variables can only have a value of True or False. ☒ True ☐ False
 - 4) A while loop in Python is a pre-tested loop. ☒ True ☐ False
 - 5) The first position in a string in Python has the index 0. ☒ True ☐ False
 - 6) The symbol '#' is used in Python to indicate a comment. ☒ True ☐ False
 - 7) "aBcD" is a string in Python. ☒ True ☐ False
 - 8) 2.3E04 is a floating number in Python ☒ True ☐ False
 - 9) The keyword **def** is used to define a function in Python ☒ True ☐ False
 - 10) The assignment operator in Python is = ☒ True ☐ False
 - 11) To divide two integers to get an integer result, you can use / True ☒ False
 - 12) The keyword **elseif** can be used in if-statements in Python. True ☒ False
 - 13) The first element of a list in Python is in position 1. True ☒ False
 - 14) In Python strings and lists are both immutable True ☒ False
 - 15) 12% 7 produces the result 5 ☒ True ☐ False
- 

Question 2: Multiple Choice - 15 Marks (1 each)

1) Which of the following statements is/are TRUE about the CPU?

- a. CPU stands for Central Processing Unit
- b. The CPU is what performs computation
- c. The CPU processes machine language
- ☒ d. Two of the above statements are true
- e. None of the above are true

2) Which of the following correctly lists the types of memory from slowest to fastest?

- a. Hard disk, Registers, RAM
- b. Hard disk, RAM, Registers
- c. External, Internal, CPU
- ☒ d. External, Internal, RAM

3) Which of the following are valid assignment statements in Python:

- ☒ a. $n = 3$
- ~~b. $x + y = 0$~~
- ~~c. $z == 3.24$~~
- d. All of the above.
- e. None of the above.

4) Which of the following adds two variables, x and y , together, divides their sum by 3 and adds 11 to the result:

- ~~a. $x + y / 3 + 11$~~
- ~~b. $(x + y + 11) / 3$~~
- ~~c. $(x + y) / (3 + 11)$~~
- ☒ d. $((x + y) / 3) + 11$
- e. None of the above are true

5) What will be the values of the variables `num1` and `num2` after the execution of the following assignments?

`num1 = 20`

`num2 = 10`

`num1 = num1 + num2 / 2` = $20 + 10/2$

`num2 = num1`

- ~~a. num1 is 20, num2 is 10~~
- ~~b. num1 is 15, num2 is 10~~
- ~~c. num1 is 25, num2 is 10~~
- ☒ d. num1 is 25, num2 is 25
- e. None of the above.

Consider the following code for the next two questions:

```
num1 = int(input("Enter a number: ")) 10
num2 = int(input("Enter a number: ")) 20
num3 = int(input("Enter a number: ")) 30
if num1 > num2 : 10 > 20 X
    if num1 > num3 :
        print(num1)
    else :
        print(num3)
else :
    if num2 > num3 : 20 > 30 X
        print(num2)
    else :
        print(num3)
```

6) Assuming that a user enters 30, 20, and 10 as the input values in that order to the code above, what is the output?

- a. 30
- b. 20
- c. 10
- d. 30 and 20
- e. Nothing, there is an error.

7) If the user enters, 10, 20, 30 as the input to the code above in that order, what is the output?:

- a. 30
- b. 20
- c. 10
- d. 10 and 30
- e. Nothing, there is an error.

8) Given the mathematics expression, $32 \leq \text{temp} \leq 100$, which of the following correctly expresses this as a Python condition?

- a. $32 \leq \text{temp} \leq 100$
- b. temp between 32 and 100
- c. $\text{temp} \leq 32$ and $\text{temp} \leq 100$
- d. $\text{temp} \leq 32$ and $\text{temp} \geq 100$
- e. $\text{temp} \geq 32$ and $\text{temp} \leq 100$

both correct

9) Consider the following code segment:

```
int total = 0;  
for x in range (1,100) :  
    total = total + x
```

Which of the following best describes what will happen if we run this code?

- a. It will sum all the integers between 1 and 100.
- ☒ b. It will sum all the integers from 1 to 99.
- c. It will sum all the odd numbers from 2 to 99.
- ~~d.~~ It will not execute the loop.
- ~~e.~~ It will enter an infinite loop.

10) What is the best term to describe the '=' character in the following code?

```
desc = "Esan is " + age;
```

- ☒ a. Equality operator
- b. Primitive type operator
- c. String operator
- ☒ d. Assignment operator
- ~~e.~~ Test operator

11) Which of the following is the index of the last element of the list marks?

- ~~a.~~ marks.len();
- ~~b.~~ length(marks);
- ~~c.~~ size(marks)-1;
- ~~d.~~ len(marks);
- ☒ e. len(marks)-1;

12) How many times does the code snippet given below display "Loop Execution"?

```
i = 1  
while i != 10 :  
    print("Loop Execution")  
    i = i + 1
```

|||| |

- a. It is an infinite loop.
- b. 0
- c. 1
- ☒ d. 9
- e. 10

13) Which of the following statements is true about functions and lists:

- ☒ a. A function can return a list.
- ☒ b. A function can be called with a list as an argument
- ☒ c. When calling a function with a list argument, the function receives a reference to the list.
- ☒ d. All of the above.
- e. None of the above. when calling a function with a list argument, the function receives a reference to the list not a copy of the list

14) What would print("%.2f" % 789.450123) output:

- a. .45
- b. 78
- c. 789.450123
- ☒ d. 789.45
- e. None of the above

789.45

15) What does the following code snippet output?

```
a=7  
b=8  
def fun(b,a):  
    a=9  
    b=8  
    return a and b
```

only
copies

} a & b are global
} this function changes their value

```
fun(a,b)  
print(a,b)
```

- ☒ a. 7 8
- b. 8 9
- ☒ c. 9 8
- d. 7 9
- e. None of the above

Question 3: Short Answers – 30 Marks (2 each, unless otherwise noted)

- 1) Consider the following code segment:

```
if a < b :  
    print("W")  
else :  
    print("X")  
    if b == c :  
        print("Y")  
    else :  
        print("Z")
```

What values of a, b and c will cause the code to print an "X" and a "Z"?

$a > b, b \neq c$

$\therefore a = 10$
 $b = 5$
 $c = 2$

- 2) Review the code snippet below:

```
maritalStatus = input("Enter your marital status (s for single, m for married): ")  
maritalStatus = maritalStatus.upper()
```

Write the statements can be used to validate whether the user entered a valid marital status?

```
if maritalStatus != "M" and maritalStatus != "S":  
    print("You did not enter a valid status.")  
else:  
    print("Thank you.")
```

- 3) What is the output of the code snippet given below?

```
s = "abcde"  
length = len(s)  
i = 1  
while i < length :  
    print(s[length - i])  
    i = i + 1
```

e
d
c
b

- 4) What does the code below compute?

```
position = 0
str = input("Enter a string")
while position < len(str) and str[position] != 'e':
    position = position + 1
print(position)
```

▷ Prompts user for a string (if the string is empty or if the first character is "e" it doesn't run the while loop)

▷ Prints the index of every character in the string, on a new line (starting at 1) and stops once the position (index number) of the string is at an "e".

- 5) The following program is supposed to sum all of the numbers entered by the user. What line of code must be inserted in the blank so that the program will achieve this goal?

```
total = 0.0
inputStr = input("Enter a value: ")
while inputStr != "":
    value = float(inputStr)
    total = total + value
    inputStr = input("Enter a value: ")
```

- 6) What is the last output line of the code snippet given below?

```
for i in range(3):
    for j in range(5):
        if i % 2 == j % 2:
            print("i") ←
        else:
            print("j")
```

Since $i=2$ and $j=4$ on the last iteration of the for loop, the last output line will be:

i

- 7) Consider the following code segment. It is designed to identify the first location within a string, `text` where two adjacent characters are the same.

```
i = 1
found = False
while not found and i < len(text) :
    _____ :
        found = True
    else :
        i = i + 1
```

What line of code should be placed in the blank to achieve this goal?

`if text[i] == text[i+1]:`

- 8) Given the following code snippet, what is final version of the list `names`?

```
names = []
names.append("Amy")
names.append("Bob")
names.append("Peg")
names[0] = "Cy"
names.insert(0, "Ravi")
```

`names = ["Ravi", "Cy", "Bob", "Peg"]`

- 9) The following code segment is supposed to display all of the elements of the list values with dashes between them. For example, if values contains [10, 20, 30, 40, 50] then the program should display 10-20-30-40-50.

```
result = ""
for i in range(len(values)):
    if i > 0:
        _____
    result = result + str(values[i])
print(result)
```

What line of code should be placed in the blank to achieve this goal?

10) What is the difference between $[7]*3$ and $7*[3]$ and what does each produce?

▷ The first copies entry 7 three times in the list: $[7, 7, 7]$

▷ The second copies entry 3 seven times in the list: $[3, 3, 3, 3, 3, 3, 3]$

11) months = ["September", "October", "November", "December"]
0 1 2/-2 3/-1

`print(months[-2][-5])`

What gets printed and why?

Python wraps around the index numbers, and negative indexes are counted from the right, starting from -1. Therefore, `months[-2][-5]` here is the same as `months[2][3]`. So the result is printed as e (months[2] is "November" and months[2][3] is "e").

12) The following code is supposed to simulate the growth of an alien population. It unfortunately is incomplete because it is missing a line of code.

```
aliens=6
def multiplyAliens():
    if aliens < 10:
        aliens = aliens*2
    else:
        aliens = aliens + 9
    return aliens
multiplyAliens()
```

What line of code needs to be introduced to make it work and where does it need to be placed?

The function is missing a "return" line. It should be placed on the last line of the function (where the dot is).

- 13) The following is incomplete fill in the blank lines to finish the code; the code is supposed to count all the 0s in a list (3 Marks).

```
def countZeroes(aList):
```

```
    count = 0
```

```
    for i in range(len(aList)):
```

```
        if aList[i] == 0:
```

```
            count = count + 1
```

```
    return count
```

2

- 14) The following code segment makes use of while statement. Convert it into code that does the same thing but makes use of the for statement. (3 Marks)

```
i = 0
```

```
j = 0
```

```
while i < 125:
```

```
    i = i + 2
```

```
    j = j + 1
```

```
print(j)
```

```
j = 0
```

```
for i in range(0, 125, 2):
```

```
    j = j + 1
```

```
print(j)
```

3

16

Question 4: Logic Errors - Correcting Code Segments - 18 Marks

- 1) The following code segment is supposed to convert a mark (an integer) to a "Pass" (60 and above), "Borderline" (50 to 59) or "Fail" (below) 50. The code as given is incorrect. Rewrite the code to fix it so that it will compute correctly. (4 Marks)

```
grade = int(input("Enter student grade: "))
if grade >= 60 :
    status = "Pass"
if grade >= 50 :
    status = "Borderline"
else :
    status = "Fail"
print(letterGrade)
```

```
grade = int(input("Enter student grade: "))
if grade >= 60:
    status = "Pass"
elif grade >= 50:
    status = "Borderline"
else:
    status = "Fail"
print(status)
```

- 2) The following code segment is to compute the difference between consecutive elements in the list `nums`. The program has three logic errors. Identify them and correct the lines of code that contain them. Note: a line may contain more than one logic error. (6 Marks)

```
nums = [12, 3, 25, 6, 11, 121, 15, 22]
for i in range(1, len(nums)):
    n = nums[i]
    m = nums[i+1]
    diff = n-m
    print("Difference is: " + str(m))
```

```
nums = [12, 3, 25, 6, 11, 121, 15, 22]
for i in range(len(nums)-1):
    n = nums[i]
    m = nums[i+1]
    diff = n-m
    print("Difference is: " + str(diff))
```

- 3) The following code is supposed to determine the positions that a string `strng` occurs in a given sentence, `sent`. The code is syntactically correct, but contains three logic errors. Identify them and correct the lines in which they occur. Note: a line may contain more than one logic error. (8 Marks)

```
sent = input("Enter a sentence: ")
strng = input("Enter a string: ")
```

0 1 2 3 4 | 5
hello
e l | 2
o .

```
i = 0
```

```
once = False
```

```
while i < len(sent) and len(strng) < len(sent): → while i < len(sent) and len(strng) < len(sent)
```

```
    found = True
```

```
    for j in range(len(sent)): → for j in range(len(strng)):
```

```
        if strng[j] != sent[i+j]:
```

```
            found = False
```

```
    if found :
```

```
        print("String " + strng + " found at position " + str(i))
```

```
        once = True
```

```
    i = i + 1
```

```
if not once :
```

```
    print("String " + strng + " does not occur.")
```


Question 6: A Little Code - 20 Marks

- 1) Write a function `nodups(lst)` that takes a list, `lst`, and returns a new list that has no duplicates. For example, if the list was `[2, 3, 4, 2, 3, 5]` then the function would return a new list `[2, 3, 4, 5]`. If the list was just `[1, 2, 3]`, then the new list would just be `[1, 2, 3]`. (10 Marks).

```
def nodups(lst):  
    lst = lst.sort() # Sorts then removes  
    for i in range(len(lst)-1):  
        while lst[i] == lst[i+1]:  
            lst.pop(i)  
    return lst
```

8
10

- 2) Write a function `sameLists(lst1, lst2)` to determine if the two lists, `lst1` and `lst2`, contain the same elements in any order. Your function should use `nodups` from previous question to first remove any duplicates from each of the lists. (10 Marks).

```
def sameLists(lst1, lst2):  
    nodups(lst1)  
    nodups(lst2)  
    isSame = False  
    for i in range(len(lst1)): # both should be same length  
        if lst1[i] == lst2[i]:  
            isSame = True  
    return isSame
```

6
10