

NEAR EAST UNIVERSITY
FACULTY OF ENGINEERING
2024-25 FALL SEMESTER
ECC102 – PROGRAMMING AND PROBLEM-SOLVING
SECOND MIDTERM EXAM

Name-Surname:	Std. No:
Department:	Signature:

- You have **100 minutes** to finish this exam.
- **WRITE CLEARLY.** We can't give you credit if we can't read and understand your answers!
- Calculators, mobile phones, laptops, tablets, or other electronic devices are **PROHIBITED**.
- A single double-sided worksheet including your name and student number is allowed.
- You have **TWO SECTIONS** in this exam.
- The first section includes **30 multiple-choice questions** (90 pts – 3pts for each).
- There is **only one correct answer** in the Section A questions.
- Multiple answers will cause getting no points from the questions.
- Mark the multiple-choice answers on the **ANSWER SHEET** provided on the next page.
- The answers on the question sheets will not be considered!
- The second section is the **code writing** section that are from your assignments (3 questions).
- The points obtained from the second section will determine your assignment points.
- There are **9 pages** in the exam paper including this page.

ANSWER SHEET FOR SECTION A.

- | | |
|-------------------------|-------------------------|
| 1. (a) (b) (c) (d) (e) | 16. (a) (b) (c) (d) (e) |
| 2. (a) (b) (c) (d) (e) | 17. (a) (b) (c) (d) (e) |
| 3. (a) (b) (c) (d) (e) | 18. (a) (b) (c) (d) (e) |
| 4. (a) (b) (c) (d) (e) | 19. (a) (b) (c) (d) (e) |
| 5. (a) (b) (c) (d) (e) | 20. (a) (b) (c) (d) (e) |
| 6. (a) (b) (c) (d) (e) | 21. (a) (b) (c) (d) (e) |
| 7. (a) (b) (c) (d) (e) | 22. (a) (b) (c) (d) (e) |
| 8. (a) (b) (c) (d) (e) | 23. (a) (b) (c) (d) (e) |
| 9. (a) (b) (c) (d) (e) | 24. (a) (b) (c) (d) (e) |
| 10. (a) (b) (c) (d) (e) | 25. (a) (b) (c) (d) (e) |
| 11. (a) (b) (c) (d) (e) | 26. (a) (b) (c) (d) (e) |
| 12. (a) (b) (c) (d) (e) | 27. (a) (b) (c) (d) (e) |
| 13. (a) (b) (c) (d) (e) | 28. (a) (b) (c) (d) (e) |
| 14. (a) (b) (c) (d) (e) | 29. (a) (b) (c) (d) (e) |
| 15. (a) (b) (c) (d) (e) | 30. (a) (b) (c) (d) (e) |

SECTION A. MULTIPLE-CHOICE QUESTIONS (90 pts – 3pts each)

1. What is the representation of the following equation in Python programming language?

$$y = \frac{b + c}{2a}$$

- ☒ a. $y = (b + c) / (2*a)$
 - b. $y = b + c / (2*a)$
 - c. $y = (b + c) / (2 \times a)$
 - d. $y = (b + c) / (2a)$
 - e. $y = b + c / 2 * a$
2. Which of the following is NOT a valid variable name in Python?
- a. CON
 - b. Con
 - c. _Con
 - d. Con_
 - ☒ e. Con-
3. Which of the following is NOT a valid variable name in Python?
- a. X1
 - b. x1
 - ☒ c. 5a
 - d. a_5
 - e. a5
4. What is the representation of the following equation in Python programming language?

$$y = \frac{b^3 + 3b}{2a}$$

- a. $y = (b ** 3) + ((3*b) / (2*a))$
- b. $y = b ** 3 + 3*b / 2*a$
- c. $y = ((b * 3) + (3 \times b)) / (2 \times a)$
- ☒ d. $y = ((b ** 3) + (3*b)) / (2*a)$
- e. $y = ((b ** 3) + (3*b)) / (a ** 2)$

Determine the outputs of the given programs in Questions 5-25.

5.

```
import numpy as np

c = np.array([[100,2,3],[10,20,30]])
print("c[0,1]")
```

- a. 100
- b. 2
- c. 3
- d. 10
- ☒ e. c[0,1]

6.

```
import numpy as np

c = np.array([[100,2,3],[10,20,30]])
a = np.array([[100,2,3],[10,20,30]])
d = a + c
print(d[0,0],d[-1,-1])
```

- a. d[0,0],d[-1,-1]
- b. 100,10
- ☒ c. 200 60
- d. [200 60]
- e. 100 30

7.

```
import numpy as np
c = np.array([100,2,3,10,20,30])
c = c.reshape(6,1)
print(c[5,0])
```

- a. Index Error
- b. [30]
- c. 2
- d. 20
- ☒ e. 30

8.

```
import numpy as np
c = np.array([[100,2,3],[10,20,30]])
counter = 0
for i in c:
    counter = counter + 1
    print(i)
    if counter == 1:
        break
```

- ☒ a. [100 2 3]
- b. 100
- c. 2
- d. 100, 2, 3
- e. 30

9.

```
import numpy as np
c = np.array([100,2,3,10,20,30])
counter = 0
for i in c:
    counter = counter + 1
    print(i)
    if counter == 1:
        break
```

- a. [100 2 3]
- ☒ b. 100
- c. 2
- d. 20
- e. 30

10.

```
import numpy as np
i = 6
c = np.array(list(range(0,i))).reshape(2,3)
print(c[0,1])
```

- a. 0
- ☒ b. 1
- c. 2
- d. 3
- e. 4

11.

```
def func(a,b,c = 5):
    return a+b+c

a,b,c = 1,5,10
x = func(a,b,c)

print(x)
```

- a. 1
- b. 5
- c. 10
- d. 11
- ☒ e. 16

12.

```
def func1(a,b):
    return a*b

def func2(a,b):
    return a+b

a,b,c = 1,5,10
x = func1(func2(a,b),b) / c

print(x)
```

- a. 6.0
- b. 5.0
- c. 30.0
- d. 10.0
- ☒ e. 3.0

13.

```
a = 3.1416
print('%0.1f' %a)
```

- a. 3
- b. 3.0
- c. 3.14
- ☒ d. 3.1
- e. 3.11

14.

```
def func1(a):
    a = a * 10

my_var = list(range(0,10,3))
func1(my_var)
print(my_var)
```

- a. [1, 4, 7]
- b. [10, 40, 70]
- ☒ c. [0, 3, 6, 9]
- d. [0, 30, 60, 90]
- e. [0, 3, 6]

15.

```
def func1(a):
    for i in range(0,len(a)):
        a[i] = a[i] * 2

my_var = list("EXAM2")
func1(my_var)
print(my_var)
```

- a. EXAM2
- b. ['E', 'X', 'A', 'M', '2']
- ☒ c. ['EE', 'XX', 'AA', 'MM', '22']
- d. EEXXAAMM22
- e. Error

16.

```
import numpy
def shape(a):
    print(a.shape)

shape(a = numpy.arange(4).reshape(2,2))
```

- a. None
- b. Error
- c. (4, 1)
- ☒ d. (2, 2)
- e. (1, 4)

17.

```
def func1(a):
    a[0] = 100
    a[-1] = 100

my_var = list(range(0,10,2))
func1(my_var)
print(my_var)
```

- a. [2, 4, 6, 8, 10]
- b. [0, 2, 4, 6, 8]
- c. [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
- ☒ d. [100, 2, 4, 6, 100]
- e. [100, 1, 2, 3, 4, 5, 6, 7, 8, 100]

18.

```
def func1(a):
    b = []
    for i in range(0,len(a)):
        b.append(a[i] * 2)

my_var = list("EXAM2")
func1(my_var)
print(my_var)
```

- a. EXAM2
- ☒ b. ['E', 'X', 'A', 'M', '2']
- c. ['EE', 'XX', 'AA', 'MM', '22']
- d. EEXXAAMM22
- e. Error

19.

```
import numpy as ecc

a = np.array([0,1,2,5])
print(a)
```

- ☒ a. [0, 1, 2, 5]
- b. 0, 1, 2, 5
- c. 8
- d. ecc
- e. Error

20.

```
import numpy as np
x = []
for i in range(4):
    x.append(i+1)

a = np.array(x).reshape(2,2)
r, c = a.shape

for i in range(r-1):
    print(a[i,-1])
```

- ☒ a. 2
- b. 1
- c. 3
- d. 4
- e. [1, 2]

21.

```
import numpy as np

x = np.array(range(0,10))
j = int(len(x)/5)
k = int(len(x)/2)
a = np.array(x).reshape(j,k)
mini = np.amin(a, 1)
maxi = np.amax(a, 1)
print(mini, maxi)
```

- a. [4 9] [0 5]
- ☒ b. [0 5] [4 9]
- c. [0 9] [1 8]
- d. [0 1 2 3 4] [5 6 7 8 9]
- e. [0 4] [5 9]

22.

```
import numpy as np

x = np.array([[0,-5,3],[-6,10,4]])
y = np.array([[2,-15,7],[4,1,6]])
print(np.amin(x,0),np.amin(y,1))
```

- a. [-5 -6] [2 -15 6]
- b. [-6 -5 3] [2 -15 6]
- ☒ c. [-6 -5 3] [-15 1]
- d. [-5 -6] [-15 1]
- e. [-5 -15 -6 0 4 1]

23.

```
res = []
for i in range(1,10,5):
    res.append(i*i)

print(res)
```

- a. [0, 25]
- b. [1, 5, 10]
- c. [0, 25, 100]
- ☒ d. [1, 36]
- e. [1, 36, 100]

24.

```
res = []
i = 0
while i < 3:
    res.append(i*i)
    i = i - 1

print(res)
```

- ☒ a. Infinite Loop (prints nothing)
- b. Infinite Loop (prints negative numbers)
- c. Infinite Loop (prints positive numbers)
- d. [0, 1, 2]
- e. [0, -1, -2]

25.

```
res = []
i = 0
while i < 3:
    res.append(i*i)
    i = i + 1
    if i**i == 9:
        break
print(res)
```

- a. Infinite Loop (prints nothing)
- b. [1, 4, 9]
- ☒ c. [0, 1, 4]
- d. [0, 1, 4, 9]
- e. [0, 1, 2]

Determine the missing statements of the given programs that produce the expected outputs in Questions 26-30.

26.

Expected Output:

```
[[ 0 -5 10  0]
 [10  4  5 11]]
```

Program:

```
import numpy as np

x = np.array([[0,-5],[10,4]])
y = np.array([[10,-0],[5,11]])
# missing statement in this line...
print(z)
```

- ☒ a. z = np.hstack((x,y))
- b. z = np.stack((x,y))
- c. z = np.vstack((x,y))
- d. x = np.hstack((x,y))
- e. y = np.hstack((x,y))

27.

Expected Output:

```
2.0
```

Program:

```
import numpy as np

x = np.array([[0],[4]])
y = np.array([[0],[11]])
# missing statement in this line...
```

- a. x = (np.mean(x) + np.min(y))
- b. y = (np.mean(x) + np.min(y))
- c. print(np.min(x) + np.min(y))
- ☒ d. print(np.mean(x) + np.min(y))
- e. print((x + y)/2)

28.

Expected Output:

```
[100, 3, 5]
```

Program:

```
def func1(a):
    # missing statement in this line...
    return a

x = [1,3,5]
b = func1(x)
print(x)
```

- a. a = 100
- b. a[1] = 100
- c. a[2] = 100
- d. a[:] = 100
- ☒ e. a[0] = 100

29.

Expected Output:

```
EVEN
EVEN
EVEN
```

Program:

```
for i in range(100):
    if i % 2 == 0:
        # missing statment in this line...
    if i == 4:
        break
```

- a. print("EVEN EVEN EVEN")
- b. print("EVEN")
print("EVEN")
print("EVEN")
- c. print("3 * EVEN")
- d.** print("EVEN")
- e. print("EVEN * 3")

30.

Expected Output:

```
aa
dd
cc
```

Program:

```
# missing statement in this line...
for i in a:
    print(2*i)
```

- a.** a = ["a", "d", "c"]
- b. a = [a, d, c]
- c. a = ["aa", "dd", "cc"]
- d. i = ["aa", "dd", "cc"]
- e. i = ["a", "d", "c"]

SECTION B

PROGRAMMING SECTION

1. Write a "Rock, Paper, Scissors" game in Python. (3 pts.)

The program will read the choices of "Player 1" and "Player 2" and print out the result (i.e., Player 1 wins. OR Player 2 wins. OR Draw).

Here are the rules of the game:

- "Scissors" beats "Paper"
- "Paper" beats "Rock"
- "Rock" beats "Scissors"

2. Write a Python function that reads two .txt files (a.txt and b.txt) and display the content of these files. (4 pts)

ANSWER:

3. Write a Python function that stores (saves) two arrays' content into two .txt files (c.txt and d.txt). (4 pts)

ANSWER: