FACULTY OF ENGINEERING
DEPARTMENT OF COMPUTER ENGINEERING

A

CENG111 / COME103 COMPUTER PROGRAMMING I
MIDTERM EXAM

Prof. Dr. M. Fatih HASOĞLU
2020-2021 Fall Semester
08.11.2021 - 13:30

— SOLUTIONS —

Student's Information

Name and Surname:	—		
Student ID:	—	Signature:	—
Exam Room:	—		

Instructions:

- Read the questions carefully and repeatedly. Writing with clarity will make it possible to assess your answers correctly.
- Any form of cheating, intention of cheating, or disruption of the exam will subject you to a disciplinary action as specified by the university's regulations.
- Carrying or using the mobile phone during the exam is not allowed. Using the phone during the exam will carry a disciplinary action.
- No food or drink to be brought into examination room.
- No student can leave the exam room during the first half hour of the exam. Any student coming after the first half hour of the exam will not be permitted to enter the exam.
- A student leaving the exam, even for legitimate reasons, may not be given a makeup exam.

Duration:	90 min	Total pages (*not* including cover page):	6	Total points:	100

# of Q/s	PO/s	Max Marks	Score
1	PO1-PO2-PO3-PO6-PO8	20	20
2	PO1-PO2-PO3-PO6-PO8	10	10
3-12	PO1-PO2-PO3-PO6-PO8	20	20
13	PO1-PO2-PO3-PO6-PO8	20	20
14-16	PO1-PO2-PO3-PO6-PO8	30	30
Total			100

Computer Programming I

CENG111/COME103 – Midterm Examination

A

Fall 2021

8 Nov 2021

Student Name: — SOLUTIONS —

Student ID:

1. TRUE / FALSE QUESTIONS   (Each 2 pts – Total 20 pts)

F In Python, math expressions are always evaluated from left to right, no matter what the operators are.

F Python formats all floating-point numbers to three decimal places by default when outputting with the print statement.

T A decision structure can be nested inside another decision structure.

T Reducing duplication of code is one of the advantages of using a loop structure.

F Both of the following for clauses would generate the same number of loop iterations.

```
for num in range(25):  → 0,24
for num in range(1, 25): → 1,24
```

F A while loop is called a *pre-test loop* because the condition is *tested after the loop* has had one iteration. ✗

T In a nested loop, the inner loop goes through all of its iterations for every single iteration of the outer loop.

F A local variable can be accessed from anywhere in the program.

T The math function `ceil(x)` returns the smallest integer that is greater than or equal to x.

F A flowchart shows the hierarchical relationships between functions in a program.

2. COMPLETION QUESTIONS: Fill in the blanks with suitable words   (Each 2 pts – Total 10 pts)

hierarchy	modulus	variable	operands	escape	comments
data types	data	counter	format	name/s	types
if	if-else	nested-if	repetition	decision	iteration
while	for	input	output	set	handler
GIGO	hierarchy	IPO	flow	accept	return
local	global	function	scope	boundary	definition
or	nor	not	nand	and	none

a) A(n) **if-else** statement will execute one block of statements if its condition is true

   or another block if its condition is false.

b) A(n) **hierarchy** chart is a visual representation of the relationships between functions.

c) A variable is accessible only to statements in the variable's **scope**.

d) The logical **not** operator reverses the truth of a Boolean expression.

e) A value-returning function has a(n) **return** statement that sends a value back to the part of the program that called it.

Student Name:

Student ID:

MULTIPLE CHOICE QUESTIONS   (Each 2 pts – Total 20 pts)

3. 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) newline
   d) separator

4. What type of loop structure repeats the code a specific number of times?
   a) condition-controlled loop
   b) number-controlled loop
   c) count-controlled loop
   d) Boolean-controlled loop

5. A value-returning function is
   a) a single statement that performs a specific task
   b) called when you want the function to stop
   c) a function that receives a value when called
   d) a function that will return a value back to the part of the program that called it

6. A variable used to keep a running total is called a(n)_____.
   a) Running total
   b) Accumulator
   c) Total
   d) Summer

7. What will be displayed after the following code is executed?
```
total = 0
for count in range(4,7):
    total += count
    print(total, end=' ')
```
   a) 4 9 15
   b) 4 9 16
   c) 5 10 17
   d) 4 5 6
   e) 5 6 7

8. What is the result of the following Boolean expression, given that $x = 5$, $y = 3$, and $z = 8$?
```
not (x < y or z > x) and y < z
```
   a) 5
   b) True
   c) False
   d) 8

Computer Programming I
CENG111/COME103 – Midterm Examination

A

Fall 2021
8 Nov 2021

Student Name:
Student ID:

9. What will display after the following code is executed?

```
def main():
    print("The answer is", magic(5))
def magic(num):
    answer = num + 2 * 10
    return answer
main()
```

   a) 25
   b) 45
   c) 55
   d) The statement will cause a syntax error.

10. Which of the following is the correct if clause to determine whether y is in the range 10 through 50, *inclusive*?

    $10 \leq y \leq 50$

   a) if 10 < y or y > 50:
   b) if y >= 10 and y <= 50:
   c) if 10 > y and y < 50:
   d) if y >= 10 or y <= 50:

11. What will be displayed after the following code is executed?

```
def main():
    num1 = 3
    num2 = 4
    answer = pass_it(num1, num2)
    print(answer)
def pass_it(x, y):
    z = x*y
    result = get_result(z)
    return(result)
def get_result(number):
    z = number + 2
    return(z)
#Calling Main
main()
```

   a) 9
   b) 12
   c) 14
   d) Nothing, this code contains a syntax error.

12. It is recommended that programmers avoid using _____ variables in a program whenever possible.

   a) local
   b) keyword
   c) global
   d) string

13. <u>ALGORITHM WORKBENCH QUESTIONS</u> **(Each 4 pts – Total 20 pts)**

a) Write Python code that inputs two integers from the user into a variables named x and y. Code should do an input validation for both numbers are positive. Otherwise it should ask to enter them again and again until both of the entered numbers are positive.

```python
x=int(input("Enter the first integer:")

y=int(input("Enter the second integer:")

while ( x <= 0 or y <= 0):

    x=int(input("Enter the first integer again:")

    y=int(input("Enter the second integer again:")
```

Note: This is accepted. One may also do the input validation for each variables. Both answers are accepted.

b) Write a function named `square` that receives one parameter (*side of square in pixels)* from the caller then it draws a square of desired size by using turtle graphics.

```python
def square ( size):

    import turtle

    for i in range(4):

        turtle.forward(size)

        turtle.left(90)
```

Note: This can be accomplished different ways, all accepted as long as square is drawn.

c) Write an if-else statement that assigns `0` to the variable `b` if the variable `a` is less than `10`. Otherwise, it should assign `99` to the variable `b`.

```python
if ( a < 10 ):
    b = 0
else :
    b = 99
```

d) Write a `for` loop that uses the `range` function to display all even numbers between -100 and 100 *inclusively* in ascending/increasing order.

```python
for i in range(-100,101,2):
    print(i)
```

e) Write a decision structure that perform the following: compares the `var1` and `var2` then display the greater of `var1` and `var2` to the screen.

```python
if ( var1 > var2 ):
    print( var1 )
else :
    print( var2 )
```

Note: This can be accomplished different ways, all accepted as long as it does the task.

Computer Programming I
CENG111/COME103 – Midterm Examination

A

HASAN KALYONCU
ÜNİVERSİTESİ

Fall 2021

Student Name: -SOLUTIONS-   8 Nov 2021
Student ID:

PROGRAMS   **(Q14-16 each is 10 pts – Total 30 pts)**

Note: Q14 and Q15 doesn't require functions but at least you must make the `main()` while writing programs. You may design Q14 and Q15 programs with functions if you would like. Answers with/without functions will be accepted for these two questions.

14. In mathematics, the notation *n*! represents the factorial of the nonnegative integer *n*. The factorial of *n* is the product of all the nonnegative integers from 1 to *n*. For example,

7! = 1 x 2 x 3 x 4 x 5 x 6 x 7 = 5040 and 4! = 1 x 2 x 3 x 4 = 24

Write a program that lets *the user enter a nonnegative integer* then uses a loop to calculate the factorial of that number. Display the factorial.

```python
# This program calculates the factorial of a positive
# integer number

#Initialize number to zero - needed for input validation
number = 0

# Get a valid number (Positive integer number) from the user.
# Using while repetition for input validation purpose
while number <= 0:
    number = int(input('Enter a nonnegative integer: '))

# Initialize the accumulator variable.
fact = 1

# Calculate the factorial of the number.
for i in range(1, number + 1):
    fact *= i     #Factorial is the product of numbers

# Display the factorial of the number.
print('The factorial of', number, 'is', fact)
```

15. Write a program that generates printable addition tests. The tests should consist of 5 questions which present a simple addition question in the following format, where the question number goes from 1 to 5, and `num1` and `num2` are randomly generated numbers between 1 and 10:

```
Question 1
num1 + num2 = _____
…
Question 5
num1 + num2 = _____
```

The program should simply display the 5 questions – it should not prompt the user for any input.

No input is received from user numbers are all randomly generated.

Example Terminal Output:

```
Question 1
6  +  5 = _____

Question 2
5  +  5 = _____

Question 3
1  +  3 = _____

Question 4
9  +  3 = _____

Question 5
6  +  1 = _____
```

```python
# This program generates printable addition tests.
# Numbers are in the range from 1 to 100.
# Number of Questions is 5
import random
def main():
    # Repating generating Questions 5 times.
    for i in range(1, 6):
        num1=random.randrange(1,11)   # Randomly 1 - 10
        num2=random.randrange(1,11)   # Randomly 1 - 10
        #Printing numbers in Question Format
        print("Question",i)
        print(num1,' + ',num2, '= _____')
        print() # Skipping one line before the next Quest.

# Call the main function.
main()
```

16. Write a program that uses turtle graphics to display a circle, square and triangle in touch with one another as shown in the below example. Main function in your program that receives the radius of circle, side of square and side of isosceles right angle triangle in pixels from the user and sent these data to functions to draw the each shapes. Beside main(), the program should also have the following functions:

`square`: Receives the side of the square from the caller and draws the square in the desired size.

`triangle`: Receives left bottom coordinates and the side of isosceles triangle and draws the triangle in the desired size and place.

`circle`: Receives the radius of circle from the caller and draws the circle in the desired size and place.

For example, if the user enters 100 for the square side, 75 for the triangle and 25 for the radius of the circle then the final shape should look as in the figure (**circle on the left, square in the middle and triangle on the right** such that they are **in contact with each** other as shown in the figure).
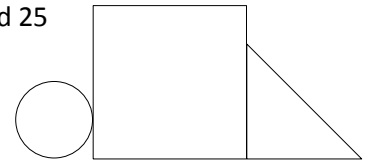
Note that it *LEFT BOTTOM CORNER OF THE SQUARE SHOULD BE* (0,0) in turtle graphics.

```python
def main():
    import turtle
    ss=int(input('Enter the side of the square: '))
    ts=int(input('Enter the side of the triangle: '))
    cr=int(input('Enter the radius of the circle: '))

# Calling function to draw the figure
    square(ss)
    triangle(ss,0,ts)
    circle(cr)
    turtle.hideturtle()
# Function Square (starting from 0,0)
def square(width):
    import turtle
    for count in range(4):      # Draw a square
        turtle.forward(width)
        turtle.left(90)
# Function Circle (starting from x=-radius, y=0)
def circle(radius):
    import turtle
    turtle.setheading(0)        # Direction is EAST
    turtle.penup()              # Raise the pen
    turtle.goto(-radius,0)      # Position the turtle
    turtle.pendown()            # Lower the pen
    turtle.circle(radius)       # Draw a circle
# Function Triangle (starting from x=ss and y=0)
def triangle(x,y,length):
    import turtle
    turtle.setheading(0)        # Direction is EAST
    turtle.penup()              # Raise the pen
    turtle.goto(x,y)            # Position the turtle
    turtle.pendown()            # Lower the pen
    turtle.forward(length)      # Draw the side
    turtle.goto(x,length)       # Draw Hypotenuse
    turtle.goto(x,0)            # Draw vertical side
# Calling Main
main()
```