

Computer Programming Final Examination

Name: _____ ID: _____

SECTION: _____ Duration: 2:30 hours Weight:40%

This is a final examination for the computer programming course.

Make sure that you have written your name, ID, and Section.

You must hold your ID while sitting on the examination or a clearance paper if you have lost your ID.

Using scientific calculators, mobile phones are **strictly** forbidden.

Mobile phones must be placed on tables found at class entrance. A student caught holding a mobile phone in his/her pocket is automatically dismissed from the exam and the university will take disciplinary decisions.

If this exam is signed by the invigilator the student will get an **F** on this course and the University will take disciplinary actions accordingly.

Make sure the total number of papers including the cover page is **5**.

You are not allowed to leave the examination room until the exam is over.

Students entering the exam room after **15** minutes are not allowed to sit on the exam.

Don't turn this page until you are instructed!

Mid Exam(30%)	Lab Assessment (30%)	Final Exam (40%)	Total (100%)

I. Code output (Pts)

```
1. rows = 6
message = "cheatingiswrong"
count = 0
for i in range(0, rows):
    print("| ",end='')
    for j in range(0, rows):
        if(i==j):
            print("\\",end=' ')
        elif(j < i):
            if(i+j > rows -1):
                print("*",end=" ")
            else:
                if(i+j == rows-1):
                    print('/',end=' ')
                else:
                    print("X", end=' ')
        else:
            print(message[count],end=' ')
            count+=1
    print("| \r")
```

```
| \ c h e a t |
| X \ i n g i |
| X X \ s w r |
| X X / \ o n |
| X / * * \ g |
| / * * * * \ |
```

```
2. i = 1
def outputX(i):
    i+=1
    print(i)
    i *= 2
for j in range(10):
    outputX(j+i)
    print(i,end=',')
    i+=2
outputX(i)
```

```
2
1,5
3,8
5,11
7,14
9,17
11,20
13,23
15,26
17,29
19,22
```

```
3. What is the problem with the following
code?
if ( True and (3.21 - 3.11 == 0.5/5)):
    print('hello')
```

The if condition is useless
Using floating point numbers for comparison

II. Code Completion (Pts)

4. Complete the following program tasked to check if two strings are balanced. For example, strings s1 and s2 are balanced if all the characters in the s1 are present in s2. The character's position doesn't matter. Eg. s1 = "Yn" and s2 = "PYnative" should return True. s1 = "Ynf" and s2 = "PYnative" should return False

```
def string_balance_test(s1, s2):  
    flag = True  
    for char in s1:  
        if char in s2:  
            continue  
        else:  
            flag = False  
    return flag
```

5. Complete the following program that accepts the name and age of a user. If the name of the user contains all vowels the program prints accepted and the persons age + 5. If the entered name doesn't contain all vowels the program prints rejected and the persons age - 5.

Example: Enter your name: Abebe	Enter your name: Eunoia
Enter your age: 21	Enter your age : 15
Output: Rejected 16	Output: Accepted 20

In the example above the name abebe contains 'a' and 'e' so it printed Rejected 16. The name Eunoia contains all vowels in the alphabet. For that reason the program printed Accepted and added 5 to the users age. Complete the following code to do the task.

```
name = input("Enter your name")  
age = int(input("Enter your age"))  
vowels = "aeiou"  
count = 0  
for i in vowels:  
    if i in name:  
        count += 1  
if(count == 5):  
    print("Accepted",age+5)  
else:  
    print("Rejected",age-5)
```

III. Code writing (Pts)

6. Write a python function that takes an integer as an argument and checks whether the given number is palindrome or not. The function must return a boolean value.

A **Palindrome** number is a number in which the reverse is the same as the string itself.

You **must** use **loops** when writing the code. **Casting the number to string is not allowed (the question won't be marked if it's done that way!)**.

For example 202 , 111, 1001.

- In the above example, you can see that 1001 is a palindrome. Because the reverse of the 1001 is 1001 itself.
- 207 is not a palindrome because the reverse is 702

```
def isPalindrome(n):
    temp=n
    rev=0
    while(n>0):
        dig=n%10
        rev=rev*10+dig
        n=n//10
    if(temp==rev):
        print("The number is a palindrome!")
    else:
        print("The number isn't a palindrome!")
```

7. Write a python function that takes two integers as an argument and checks whether the numbers are Amicable or not. The function must return a boolean value.

Amicable numbers are two different numbers so related that the sum of the proper divisor of each is equal to the other number. In other words, the sum of all proper divisors of number1 and the sum of all proper divisors number2 should be equal.

```
1. x=int(input('Enter first number : '))
2. y=int(input('Enter second number : '))
3. sum1=0
4. sum2=0
5. for i in range(1,x):
6.     if x%i==0:
7.         sum1+=i
8. for j in range(1,y):
9.     if y%j==0:
10.        sum2+=j
11. if(sum1==y and sum2==x):
12.     print("Given numbers are Amicable!")
13. else:
14.     print("Given numbers are not Amicable!")
```