National University of Computer and Emerging Sciences

School of Computing **Fall 2015** **Islamabad Campus**

CS101
Introduction to Computing

Monday, September 14, 2015

Course Instructor

Dr. Sibt ul Hussain, Dr. Abul Malik,
Ms. Uzma Maroof

Serial No:

Mid I

Total Time: 1 Hour

Total Marks: 80

_____
Signature of Invigilator

_____ _____ _____ _____

Student Name Roll No Section Signature

DO NOT OPEN THE QUESTION BOOK OR START UNTIL INSTRUCTED.

Instructions:

1. Attempt on question paper. Attempt all of them. Read the question carefully, understand the question, and then attempt it.
2. Please read the complete paper before attempting any question and manage your time intelligently.
3. No additional sheet will be provided for rough work. Use the back of the pages for rough work.
4. If you need more space write on the back side of the paper and clearly mark question and part number etc.
5. After asked to commence the exam, please verify that you have ten (**10**) different printed pages including this title page. There are total of **5** questions.
6. Use permanent ink pens only. Any part done using soft pencil will not be marked and cannot be claimed for rechecking. Make a smiley on front page and earn four bonus marks.
7. Use **proper indentation** while writing code and make sure that your code is legible. Failing to do so can cost you marks.

	Q-1	Q-2	Q-3	Q-4	Q-5	**Total**
Marks Obtained						
Total Marks	25	25	10	10	10	**80**

Vetted By: _____Vetter Signature: _____

25

Q. No. 1

(a). For each expression at left, indicate its value in the right column. List a value of appropriate type. e.g., 7 for an integer, 7.0 for a real, "hello" for a String,

10

Expression	Value
-(6 + 3 - 2 * 3)	
15 % 6 + 5 % 5 + 12 % 7 % 3	
9 / 2 / 2.0 + 9 / 2.0 / 2	
5 + 6 + 7 % 82**3 + 9 + 2 * 3**	
not(3 < 7 and -1 != 8)	
31 / 2 / 10.0 + 10 / (5 / 2.0)	
5 >= 5 * 6 + 2 or 9 > 4 * 5	
(2.5 = 5 / 2.0) * 10	

True or Talse for a boolean or write error if there is any.

(b) For each run and input below, write the output that is produced.

5

```
y =0
n = input("Enter an Integer: ")
while n!=0:
      x=n%10
      y=y*10+x
      n=n/10
print(y)
```

Dry run:

Run 1
Enter an Integer: 452
Output:

Run 2:
Enter an Integer: 1462
Output:

(c) What is the output of the following pseudo code:

5

```
a=input('Enter a Number: ')
b=input('Enter another Number: ')
if a % 2 != 0:
          a = a * 2

if a > 10 :
          b=b+1;
elif a < 10:
          a=a-1
          b=b-1;

print(str(a)+" "+str(b))
```

Dry run:

Runs	Outputs
Input for Run1: Enter a Number: 12 Enter another Number: 12	**Output of Run1:**
Input for Run2: Enter a Number: 7	**Output of Run2:**

Enter another Number: 4	
Input for Run3: Enter a Number: 5 Enter another Number: 8	**Output of Run3:**
Input for Run4: Enter a Number: 3 Enter another Number: 42	**Output of Run4:**

(d) What is the output of the following pseudocode:


```
a=input('Enter a Number')
b = 1
c = 0
while b <= a:
     if b % 2 == 0:
          c=c+1

     b = b * 10
     c=c+1

print( c + 1)
```

Runs	Outputs
Input for Run1: Enter a Number: 4097	**Output of Run1:**
Input for Run2: Enter a Number:388	**Output of Run2:**

25

Q. No. 2

a) Write an if statement for the following situation: If an integer variable `currentNumber` is multiple of 5, change its value so that it is now 3 times `currentNumber` plus 1, otherwise change its value so that it is now half of `currentNumber`.

3

```
if currentNumber % 5 == 0:
   currentNumber = currentNumber*3 +1
else:
   currentNumber = currentNumber/2
```

2

b) Write an if/else statement to check the input given by the user in variable `city` is "Lahore" otherwise show him message that his input is wrong.

```
if city != "Lahore":
    print "Your input is wrong"
```

c) Write an if statement that assigns "High" to variable `x` if the variable *`grade`* is not in the following ranges [-10, 10] and [100, 1000] (inclusive).

2

```
if not (-10 <= grade  <= 10) and not (100 <= grade <= 1000):
   x = "High"
```

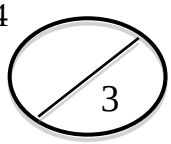
d) Write a while loop that displays sum of the first `n` odd numbers:

5

1 + 3 + 5 + . . . + 2n − 1 …

```
sum = 0
counter = 1
NextNumber = 1
while counter <= n:
   sum = sum + NextNumber
   NextNumber = NextNumber + 2
   counter = counter + 1
```

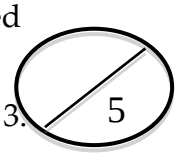
e) Write a while loop that displays the following sequence: 1, 4, 9, 16, 25, 36, 49, 64




```
counter = 1

while counter <= 8:
   print (counter**2 ,)
   counter = counter + 1
```

f) Write a while loop that finds number of times a whole number `n` can be divided by 4 (using integer division) before reaching 1. For instance, if user gives 10 as input the result should be 1 and if a user gives 64 as input the result should be 3.




```
n = input()
counter = 0
while n >= 4:
    n =n//4
    counter = counter + 1
print counter
```

g) What will be the output of following statements, write errors if they results in syntax error. Please write your output in the second column.

5

Code	Output
x= '5' y=10 print(x+y)	Error Line 3: Can't convert 'int' to str
True=5 z=20 print (True / z)	Error Line 1 : Assignment to Keyword
w=---5 q=20 print (-w*+q)	100
a=50 b=100 print(str(b%a)+" "+Str(a%400)+" ")	Error: name 'Str' is not defined
a=5*8 b=8*5 e=a+b f=e/4 print(e)	80

Q. No. 3 The Syracuse (also called Collatz or Hailstone) sequence is generated by starting with a natural number (integer) and repeatedly applying the following function until reaching 1:

10

$$syr(x) = \begin{cases} x/2 & \text{if x is even} \\ 3x+1 & \text{if x is odd} \end{cases}$$

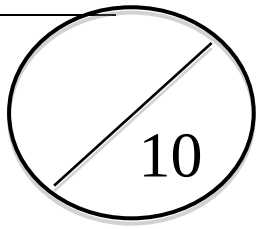
For example, the Syracuse sequence starting with 5 is: 5, 16, 8, 4, 2, 1.

Now write a program that gets a starting value from the user and then prints the Syracuse sequence for that starting value.

```
x = int(input())

while x != 1:
    print (x)
    if x%2 == 0:
        x = x/ 2
    else:
        x = 3*x + 1
print (x)
```

Q. No. 4 Write the code (using a loop) that takes size of square as input and prints a square of that size using '*' characters. Note you are only allowed to use a single loop.



For example:

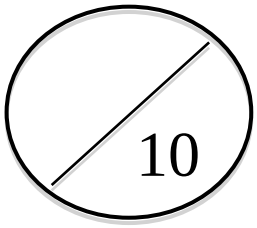
```
Please Enter size of Square: 7
* * * * * * *
*           *
*           *
*           *
*           *
*           *
* * * * * * *
```

```
Please Enter size of Square: 5
* * * * *
*       *
*       *
*       *
* * * * *
```

```
Size = input()
print ('*'*Size)
line = 1
while line <= Size-2:
    print ('*'),
    print(' '*(Size-2)),
    print ('*')
    line = line + 1
print ('*'*Size)
```

Solution 2:

```
x=input('Please Enter size of Square: ')
i=1
while i <= x:
     if i==1 or i==x:
          print x*'* '
     else:
          print '* '+' '*(2*(x-2)-1)+' * '
     i=i+1
```



Q. No. 5 Write a program that accepts four integers `hour1, minute1, hour2, and minute2` as inputs. Each pair of inputs represents a time on the 24-hour clock (for example, 1:36 PM would be represented as 13 and 36). The program should print "you can take lunch" if the gap between the two times is long enough to eat lunch: that is, if the second time is at least 45 minutes after the first time. Otherwise the program should print "you can take your lunch"

You may assume that all parameter values are valid: the hours are both between 0 and 23, and the minute parameters are between 0 and 59. You may also assume that both times represent times in the same day, e.g. the first time won't represent a time today while the second time represents a time tomorrow. Note that the second time might be earlier than the first time; in such a case, your program should print "Invalid Input".

Sample Run of Your Program:

```
Enter First Hour: 11
Enter First Minutes: 59
Enter Second Hour: 12
Enter Second Minutes: 0

You cannot take your lunch
```

Solution:

```
hr1=input('Enter First Hour: ')
min1=input('Enter First Minutes: ')
hr2=input('Enter Second Hour: ')
min2=input('Enter Second Minutes: ')
time1=hr1*60+min1
time2=hr2*60+min2
print('\n')
if time1 > time2:
     print('Invalide input')
elif time2-time1 >=45:
     print ("you can take your lunch")
else:
```

```
print ("you cannot take your lunch")
```