Indian Institute of Technology Gandhinagar

ES 112: Computing

Mid-Semester Examination | Total Marks: **100**

Date: 20 Sep 2014 | Time: **2 hrs.**

Your info

Roll number:

Name:

Section:

Your Score

Q1	Q2	Q3	Total

Instructions :

- Write your roll number and name clearly, failing which you will lose points.
- You have to answer in the question paper itself in the given area. You will be given extra papers for rough work, which should be attached with the main sheet.
- Grading will be done based only on the answers on the main question paper.
- Do not overwrite. Unclear/Ambiguous answers may be discarded.

---

1. **Multiple Choice Questions**. Circle the answers, note that there might be multiple correct answers to the same question. No need to give the reason. $10 \times 2 = 20$**.**

   (a) Which statements are correct python syntax?

   i. `x != x`
   ii. `False = 0`
   iii. `x + 2 = 1`
   iv. `print this value`

   (b) Which are correct input statements?

   i. `x = input()`
   ii. `x = input "enter something:"`
   iii. `x = input( enter a number )`
   iv. `input() = x`

   (c) Which print statements are correct?

   i. `print "9" + "9"`
   ii. `print "is this a string`
   iii. `print 9 + 9`
   iv.
   ```
   nothing = 1
   print nothing
   ```

   (d) Which are correct python expressions?

   i. `2 == 1 + 1`
   ii. `2 = 1 + 1`
   iii. `seven = 3 * 4`
   iv. `x = 2(3 + 3)`

   (e) Which operations result in 8?

i. `65 / 8`
ii. `17 % 9`
iii. `2 ** 4`
iv. `64 ** 0.5`

(f) Which are correct python statements?

i. while a = 1:
ii. `while a+7:`
iii. `while len(c)>10:`
iv. `while a and (b-2 == c):`

(g) Which are correct type conversions?

i. `int (7.0 + 0.1)`
ii. `float(9 / 0 )`
iii. `int("eleven")`
iv. `int (float ( 2) )`

(h) Which python mathematical expressions are correct?

i. `'third'*'a charm'`
ii. `'eggs'/ 2.0`
iii. `'13.0' - '0'`
iv. `'12' * 0`

(i) Suppose we define the following function:

```
def fun(a, b):
   print a + float(b)
```

Which of the following is/are not a valid call(s) to `fun(..)` :

i. `fun(1, 2)`
ii. `fun(2, "4")`
iii. `fun("4", 5)`
iv. `fun("4", "4")`

(j) Which python statement(s) give you an error, either due to syntax, or division by zero or some other illegal operation?

i.
```
print "seven !  = eight, True or False ", seven == eight
```

ii.
```
x = 100
x / ( 1 / 2 * 2 / 4  - 1 / 2 * 2 / 4.0 )
```

iii.
```
x = 10
if (x <= 1):
   print "less than one"
   elif x > 1:
      print "greater than one"
      else:
        print " nothing"
```

iv.
```
def addone_to_input(x):
   return 1
addone_to_input(2)
```

2. **For each program below, write the outputs that you will get when you run them. If it produces an error at some point in the execution write that down, and say why. If the program produces some output and then gives an error, write down the output and then write error and say why. You do not have to reproduce the exact error message that python gives out, but you need to point out why the error is occurring.** $10 \times 4 = 40$.

(a)
```
print type(1/2)
print type(1.0/2)
print type("1" * int("2")  + int("1")* "2" )
a = 10
print "printing a ", a
print ("this is a = "  + a)
```

(b) The following output was obtained from python console. Using this information, find out the precedence of `%` with respect to `*` among the three options :

- higher than multiplication
- lower than multiplication
- same as multiplication, evaluated left to right

```
>>>print   25 * 3 % 4 * 4 % 3
   0
```

(c)
```
x = input()
while True:
   if x * x >= x:
      x = x + 1
   else:
      break
print x
```

(d)
```
x = 234
while x %10 != 0:
   x = x + 1
print x
```

(e)
```
x = 10
for i in range(2, x/2):
    x = x + i
print x
```

(f) Write down the output when i) x = 5, y = 2 and when ii) x = 0, y = 4.

```
x = input()
y = input()

x = x * y
```

```
y = x / y
x = x / y

print "x = "
print "y = "
```

(g)
```
counter = 1
def doLotsOfStuff():
    global counter
    for i in [1, 2, 3]:
        counter = counter + i

doLotsOfStuff()
print counter
```

(h) Write down the output when `x=10, y = 21`.

```
x = input()
y = input()

if (x + y >= 31):
   print "the first if condition is true"
elif (y % x == 1):
   print "the second condition also is true"
else :
   print "the third option"

print "Finally"
```

(i) Write down the output i) when `x = 0.5` and ii) when `x = "2.5"`?

```
def func1(y):
  y = float(y)
  if y <= 1:
     x = 1/y
  else:
     x = y
  return x

def func2(x):
   x = float(x)
   if x < 2:
       y = 1/x
   else:
      y = x
   return y

x = input()
```

```
print func1(func2(x))
```

(j) Write down the output when `a = 10, b = 20`.

```
a =  input("a = ")
b = input("b = ")

def func1(a):
   a = a + b
   return a

def func2(b):
   b = b - a
   return b

a = func1 (b)
b = func2 (a)
print "a = ", a
print "b = ", b
```

3. **Write down a program for each of the following questions. For each question create at least one appropriate function.**: 4×10.

   (a) Write down a program to do the following– it should ask for first and last names of users, the month, the day and the year and will output a sentence, saying when the user was born and whether it was a leap year. Also print out a sentence saying which year they will be 100 years old. An example input/output is given below.

   Hint: Leap year is when the year is divisible by 4 normally. However, if the year is divisible by 100, e.g. 2000, then the year has to be divisible by 400 to be a leap year.

```
Enter your first name: Chuck
Enter your last name: Norris
Enter your date of birth:
Month? March
Day?10
Year?1940
Chuck Norris was born on March 10, 1940, which is a leap year.
Chuck Norris will be 100 years old in 2040.
```

(b) Take a number n, write down a program that returns `True/False` based on whether n satisfies the Pythagoras rule, i.e. there are two numbers a and b, both smaller than n, such that

$$a^2 + b^2 = n^2.$$

Hint: While it is fine if you give a solution with nested for loops, this can be done with no nesting of for loops. Feel free to use the `math` library.

(c) Input n, then find the sum of all the multiples of 3 or 5 that are (strictly) smaller than n. If we list all the natural numbers below 10 that are multiples of 3 or 5, we get 3, 5, 6 and 9. The sum of these multiples is 23.

(d) Write a function that creates guessing game between the user and the computer. The function first stores a number between 1 and 10000 in a variable `x` by executing `x = random.randint(1, 10000)`, (remembering to `import random`). Then onwards, the function asks the user for a guess of the number and reports `High` or `Low` depending on whether the user's guess is higher or lower than the value in `x`. The function returns only if the user guesses correctly.