4

Time Allowed: 3 Hours

MCA-101 Object Oriented Programming

Master of Computer Applications
Semester I, December 2017

M.M. 70

1. Give the output produced by the following code (Assume that function IDs are generated in the sequence 1001, 1002, ...): 5

```
def f():
    print('global f')

def g():
    global f
    print ('id(f): ', id(f))
    def f():
        print('inner f')
    print('id(f): ', id(f))
    f()

def h():
    print('inside h: id(f): ', id(f))

def main():
    print(id(f))
    f()
    g()
    g()
    h()

if __name__ == '__main__':
    main()
```

2. Write a recursive function computeValue which takes inputs d and n and returns the computed value as shown below: 10

```
Input: d (digit from 1-9) and n (number of times)
Returns: The value (d + dd + ddd + d…d n-times)
Example:
Input: for d = 9 and n=3
Output: 9 + 99 + 999 = 1107
```

Show the contents of the run-time stack for each call to function computeValue for the inputs d = 9 and n=3.

3. Write a function that takes three lists of strings and produces output as below. 10

```
Input: Lists of subject, verb, and object
Output: list of all the possible statements of three words where word1
is from subject list, word2 is from verb list, and word3 is from object
list, followed by a full stop.
```

else
~~int(str(d)*n)+~~

```
example:

Input lists:
```

9+99
105

```
Subject = ['I', 'You']
verb = ['like', 'love']
object = ['football', 'programming']

Output:
['I like football.',  'I like programming.', 'I love football.',
'I love programming.', 'You like football.', 'You like programming.',
'You love football.', 'You love programming.']
```

4. Write a recursive function to combine two dictionaries by combining the values for common keys in the form of a list. 10

```
Example:
d1 =  {'a':100, 'b':200, 'c':300}
d2 =  {'a':300, 'b':400, 'd':400}
returns d3 = {'a':[100,300], 'b':[200, 400], 'c':300, 'd':400 }
```

5. Write a recursive function to compute sum of a nested list, nested up to any level. 10

```
Example:
Input list = [1, 4, [5, 3], [2,[3, [4]]]]
Return value = 22
```

6. Create a class `Person` whose objects comprise of two attributes: `name`, `DOB`. Create a class `Student` that inherits from the class `Student` and includes additional attributes `rollNo`, `marks`. Write a python program to read a file `student` of objects of type `Student` and produce an output file by increasing the `marks` of each student by `5`, In case, the modified value exceeds `100`, use `100` as `marks`. You should include the necessary checks for exception handling. 10

7. In each case give the output produced, if any. Also, point out errors if any: 15

i)

```
class A:
    pass

class B(A):
    pass

print(isinstance(A(), A))
print(type(A()) == A)
print(isinstance(B(), A))
print(type(B()) == A)
```

ii)

```
spam = 'helloSpamSpamEggSampSampS'
print(spam.strip('ampS'))
print(spam.strip('Spam'))
print(spam.strip())
```

iii)

```
a = [1, 2, 3]
b = [2, 1, 3]
print(id(a) == id(b))
c  = {1:1, 2: 4, 3:9}
d = {1:1, 3:9, 2: 4}
print(c == d)
```

iv)

```
def f(a = []):
    a.append([1, 2, 3])
    return a

print(f())

def f(a = []):
    a.extend([1, 2, 3])
    return a
print(f())
```

v)

```
x = 'abc'
def f():
    x = 'def'
    print('x = ', x)
f()
```

vi)

```
 x = 'abc'
def f():
    x = x + 'abc'
    print('x = ', x)
f()
```