The University of Melbourne
Department of Computing and Information Systems

ISYS90088
Introduction to Application Development
Semester 2, 2016

Identical examination papers: None

Exam duration: Three hours

Reading time: Fifteen minutes

Length: This paper has ***four* (4)** pages including this cover page.

No. of sections: Three

Authorized materials: No materials are authorized. Calculators are *not* permitted.

Instructions to invigilators: Students must not remove any part of the examination paper from the examination room. Students should be supplied with the exam paper and a script book, and with additional script books on request.

Instructions to students:

- This paper counts for 60% of your final grade.
- *All questions should be answered* by writing a brief response or explanation on the lined pages in the script book. The reverse side of any page may be used to make rough notes, or prepare draft answers.
- Use a blue or black pen (not a pencil).
- Please write your student ID below and on your script book.
- When you are finished, place the exam paper inside the front cover of the script book.

Library: This paper will *not* be held by the Baillieu library.

Student's ID number:

Examiners' use only:

Total [60]

Section A: [15 marks] – Answer all questions

1. **[1 * 10 = 10 marks]** Evaluate the following expressions, and provide the output in each case:

```
a. ('ISYS', 90087 + 1)

b. sorted({'dogs': ['Anne'], 'ducks': ['Mathew']})[-1]

c. "ISYS9" + str(0)*2 + "Application"[:1][0]

d. bool(3 > 0 and len([]))

e. (1, (2, 3), 4)[1][0]

f. 7 // 3

g. ("pi = {0:10.5f} != {1:.5f}".format(3.141592653, 22/7))

h. 1 == 1 or 2 > 1 and 1 > 2

i. ('exam', 'Example', 'Ex!')[1:]

j. {'name': 'kev', 'Age': 25, 'ID': 2314, 'pay':
   24.99}.pop('gender', 'not found')
```

2. **[5* 1 = 5 marks]** State whether the following statements are true or false:

a. The `float()` function can be used to convert a `float` value to an `int`
b. A statement in one function can access a local variable in another function
c. A `return` statement returns a value back to the part of the program that called it
d. When an input file is opened, its read position is initially set to the first character in the file
e. The file method `readline()` is used to read the entire contents of a file in a single operation

Section B: [20 marks] – Answer all questions.

3. **[2 marks]** What is the value of `total` after the following code has been executed?

```
i = 0
total = 0
while i < 9:
    if i%4 == 0:
       total = total + i
    i = i + 2
```

4. [**3 marks]** What will the following program print?

```
def main():
    x = 1
    y = 3.4
    print(x, y)
    change_us(x, y)
    print(x, y)

def change_us(a, b):
    a = 0
    b = 0
    print(a, b)

main()
```

5. **[4 marks]** Rewrite the following function, replacing the while loop with a for loop, but preserving the remainder of the original code structure:

```
def get_even(nums):
   evens = []
   i = 0
   while i < len(nums):
      num = nums[i]
      if not(num % 2):
         evens.append(num)
      i += 1
   return evens
```

6. **[6 marks]** The following function is meant to take a dictionary as input, and return a dictionary containing only those (key, value) pairs in the original dictionary where the key is the same as the value. However, in its current form, the code contains a number of errors. Identify any 3 errors in the code (using the provided line numbers), determine for each whether it is a "syntax", "run-time" or "logic" error, and provide a replacement line which corrects the error.

```
1 def same_kv(dic):
2    out = []
3    for key,val in d.items():
4       if key = val:
5          out[val] = key
6    print out
```

7. **[5 marks]** What is the output of the following code:

```
wlist = [5,10,4,2]
daily_limit = 3
transactions = 0
balance = 12
while balance and transactions < daily_limit:
   balance -= wlist.pop()
   transactions += 1
   print (transactions,balance,wlist)
```

Section C: [25 marks] – Answer all_questions

8. **[5 marks]** There are three seating categories at the Melbourne Cricket Ground (MCG) stadium. For an Australian Football League (AFL) game, Class A seats cost $100, Class B seats cost $50, and Class C seats costs $25. Write a program that asks how many tickets were sold for each class of seats, and then displays the amount of income generated from ticket sales. Make sure to provide appropriate comments. For example:

```
How many tickets were sold for Class A seats? 2000
How many tickets were sold for Class B seats? 5000
How many tickets were sold for Class C seats? 30000
The total amount generated from tickets sales was: 1200000
```

9. **[8 marks]** Write a function called `minSum`, which takes as an argument a list of lists of numbers, calculates the sum of the values in each of the lists and returns the smallest such sum. Assume the list is a non-empty list. For example, if the argument was: [[1, 4, 0, 2], [2, 5, 1, 3], [3, 6, 2, 4]], the sums would be 7, 11, and 15, so the value returned should be 7. Make sure to provide appropriate comments.

10. **[12 marks]** A "palindrome" is a non-empty string which has the same sequence of letters when read from the front or from the back of the word. For example, `"abba"` and `"glenelg"` are palindromes. A "multi-palindrome" is a word, which is made up of two or more repeats of a palindrome. For example, `"azaaza"` is a multi-palindrome because it contains the palindrome `"aza"` repeated twice. Write the function `multipal` which takes a string argument `word` and an optional argument `n` in the form of a positive integer `(n ≥ 2)`, and returns `True` if word is a multi-palindrome, and `False` if not. If a value is supplied for the optional argument `n`, then the function should return `True` if and only if `word` is made up of exactly `n` repeats of a palindrome. You may assume that all arguments to `multipal` are valid.
For example:

```
>>> multipal('abaaba')
True
>>> multipal('abaabax')
False
>>> multipal('glenelg')
False
>>> multipal('abaaba',2)
True
>>> multipal('abaaba',3)
False
>>> multipal('aaaa',4)
True
>>> multipal('')
False
```