



University College Dublin
An Coláiste Ollscoile, Baile Átha Cliath

SEMESTER II EXAMINATION – 2016/2017

COMP 10020

Introduction to Programming 2

Prof. S. Dobson

Prof. P. Cunningham

Dr. B. Mac Namee *

Time Allowed: 1 Hour

Instructions for Candidates

Answer Question 1 and any two other questions.

Total marks available **100**. The value of each part of each question is shown in brackets next to it.

This is a Closed Book/Notes exam. Students are **not** permitted to bring materials to the Exam Hall.

Instructions for invigilators

This is a Closed Book/Notes exam.

Students are **not** permitted to bring materials to the Exam Hall.

Students are **not** permitted to use a computational device or the Internet in any fashion.

1. (a) What is the output of `print str[3:8]` if `str = 'Hello World!'`? (4)

(b) Only one of the following Python variable names is valid:

- `3rd_degree`
- `__d`
- `my word`
- `for`

Identify the valid variable name, and explain what is wrong with each of the others.

(4)

(c) Suppose the following code is used to create a dictionary:

```
d = {"leinster":40, "munster":45}
```

What happens when the following code is executed:

```
d["ulster"] = 17
```

(4)

(d) What is output by the following Python code snippet?

```
i = 0
while i < 10:
    print( i )
    i += 2
    if i == 6:
        break
print(0)
```

(4)

(e) What is the difference between a *tuple* and a *list* in Python?

(4)

(f) What value is printed after the following code is executed?

```
a = b = 1
b += a + a * 8
print(b)
```

(4)

- (g) The following code snippet defines the start of a class to represent account holders at a bank:

```
class AccountHolder:  
  
    def __init__(self, name, number):  
        self.name = name  
        self.number = number  
        member_count = 1  
        testVal = 14
```

Name the *instance variables* in the code given above.

(4)

- (h) Write a **toString** method for the class defined in part (g) that returns a string containing the account holder's name and number as follows:

A/c holder: "Mike" [716453]

Where the club member's name is **Mike** and account number is **716453**.

(4)

- (i) Fill in the code to complete the following function for computing the *factorial* of an integer.

Note: The factorial of an integer is the product of the integer and all integers below it; e.g. factorial four ($4!$) is equal to $4*3*2*1 = 24$ and factorial six ($6!$) is equal to $6*5*4*3*2*1 = 720$

```
def factorial(n):  
    if n == 0:  
        return 1  
    else:  
        _____ # Complete this line
```

(4)

- (j) Suppose that

s1 = {1, 2, 4, 3}

and

s2 = {1, 5, 4, 13}

What is the value of the expression **s1 & s2** ?

(4)

2. The following Python code declares a class used to represent an athlete object.

```
# A class to represent a rugby player
class RugbyPlayer:
    # Constructor
    def __init__(self, name, dob, club, position):

        # Name stored as a string
        self.name = name
        # Date of birth string in the form dd/mm/yyyy
        self.dob = dob
        # Club stored as a string
        self.club = club
        # The player's main position stored as a string
        self.position = position

    # A method to print the details of the rugby player
    def show(self):
        print(name + " (" + dob + "), " + club
              + ", " + position)
```

- (a) Write Python code to create two rugby players with the following attribute values:

Name:	Rory	Name:	Niamh
Club:	Ulster	Club:	Munster
Dob:	15/08/1982	Dob:	30/09/1984
Position:	Hooker	Position:	Full-Back

Also call the **show** method for each object created.

(10)

- (b) Although the class definition code written to answer part (a) will run, an error of type **NameError** will occur when the **show** method is called. Write a new version of the **show** method that corrects the error.

(5)

Question continues on next page...

(c) Write a new Python class to represent professional rugby players that *inherits* from the **RugbyPlayer** class. The **ProRugbyPlayer** class should have the following characteristics:

- Include an extra *instance variable* called **salary** that stores the player's annual salary.
- Include an extra *instance variable* called **weights** that stores measures of the player's weight made at particular dates. Weights should be stored in an appropriate collection data structure that allows storage of both the date on which a measure was made and the weight statistic itself.
- To initialise a **ProRugbyPlayer** object, the name, date of birth, club, position, and salary attributes should be initialised with values supplied as parameters and an empty collection of weight statistics should be created.
- Include a method **addWeightStat** that adds a weight statistic (in kilograms) to the collection of weights. Both the date the measure was made and the weight statistic should be provided as parameters to this method.
- Include a **show** method that performs the same operation as the **show** method from the **RugbyPlayer** class but also prints the list of weight statistics.

(15)

3. (a) Describe using pseudo code, and diagrams where appropriate, how the **insertion sort** algorithm works.

(15)

(b) Show the steps the bubble sort algorithm would take to sort the following list:

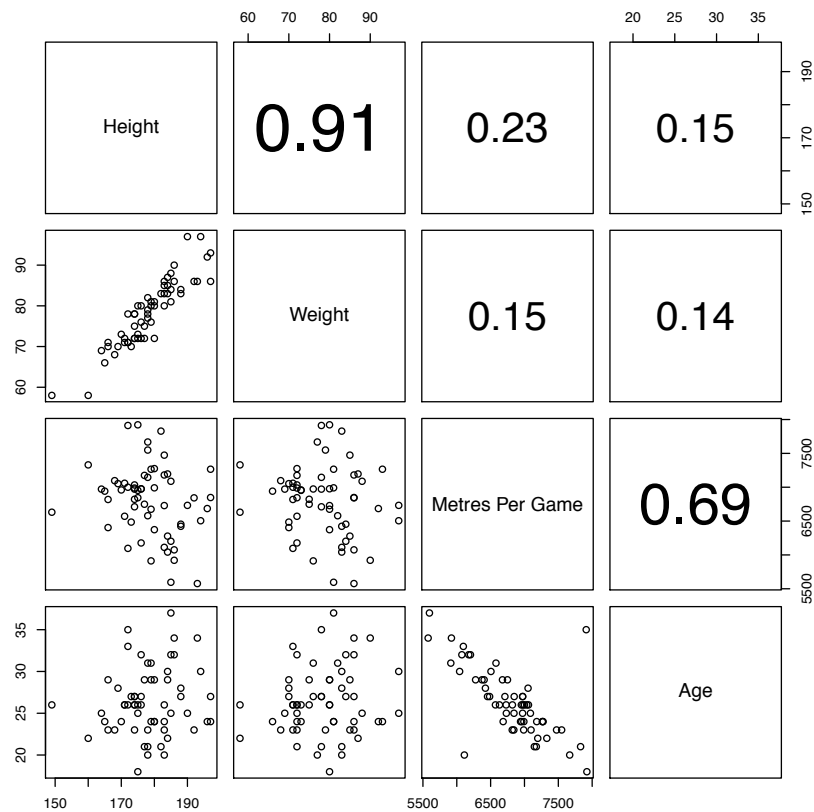
[49, 21, 88, 11, 66]

(10)

(c) A colleague has suggested to you that *bubble sort* is a faster approach to sorting than *insertion sort*. Is your colleague correct? Why or why not?

(5)

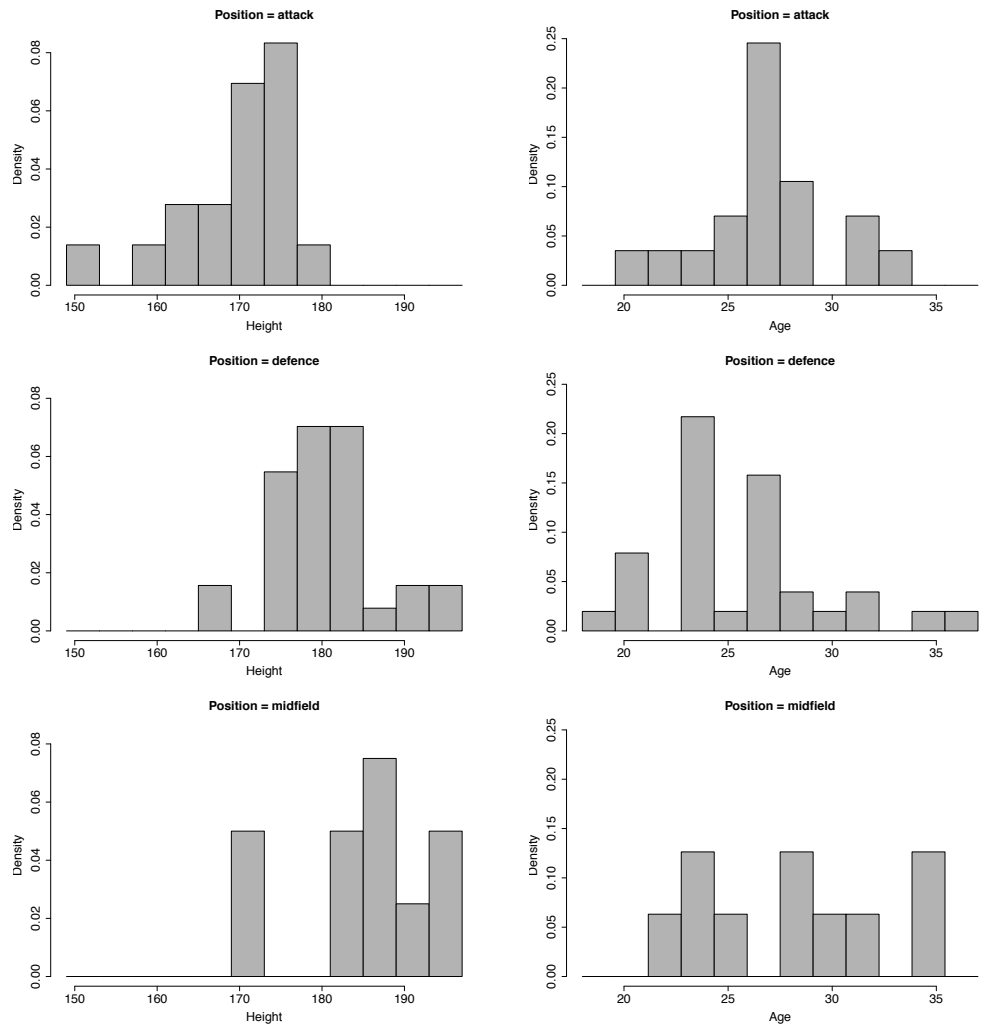
4. (a) (i) The image below shows a **scatter plot matrix** (SPLOM) of a dataset that describes the players on a soccer team. The variables included are players' height, weight, and age; and the average number of metres run by a player per game. Describe the relationships between these four variables that are apparent from this data visualisation.



(10)

Question continues on next page...

- (ii) The images below show sets of **small multiple histograms** that illustrate (a) player height broken down by player position, and (b) player age broken down by player position. What can be learned about the relationships between position and player height and position and player age from these data visualisations?



(5)

Question continues on next page...

- (b) The following simple **pandas data frame** is stored in the Python object **countries**.

Country	GDP	LifeExp	Population
China	11384760	75.41	1357000000
India	2182580	68.13	1252000000
United States	17968200	79.68	321068000
Indonesia	888648	72.45	249900000
Brazil	1799610	73.53	200400000
Pakistan	246849	67.39	191854000

Write Python code (using the pandas library) to perform the following operations:

- (i) Calculate the **maximum** value for each column in the data frame. (3)
- (ii) Select only the rows from the data frame with a population **greater than or equal to 300,000,000**. (4)
- (iii) Select only the **GDP and LifeExp** columns from the data frame. (4)
- (iv) Create a new column in the data frame called **Value** that contains the value of **GDP** multiplied by **Population** for each country. (4)