Semester 1 Examinations 2018 / 2019

Exam Code(s)	1CSD1, 1CSD2
Exam(s)	M.Sc. in Computer Science (Data Analytics)
Module Code(s)	CT5102
Module(s)	Programming for Data Analytics
Paper No.	I
External Examiner(s)	Professor P. L. Lanzi
Internal Examiner(s)	Professor Michael Madden *Dr. Jim Duggan

Instructions: Answer any 3 questions. All questions carry equal marks.

Duration	2hrs
No. of Pages	5 (including cover page)
Department(s)	Information Technology

Requirements

1. (a) Consider the following code snippet:

```
e    <- new.env()
e$f1 <- function()1
y    <- e$f1()
```

Use this example to explain the difference between an *enclosing environment* and a *binding environment*. Use a diagram to show the variables, function and their environments.

[4]

(b) What does the following code return? Explain the mechanism by which the value is calculated.

```
f2 <- function (x){
  function(y){
    x-y
  }
}

f2(5)(10)
```

[4]

(c) Draw a diagram of the global environment after the following code has been run. Explain the significance of the <<- operator.

```
y <- 100
f3 <- function (x){
  z <<-200
  y <-x-1
}

ans <- f3(10)
```

[3]

(d) Write a function ***mystack*** that returns a list of functions (a closure) to manipulate the stack vector. The functions are:

Function Name	**Details**
push(v)	Push a value onto the stack
peek()	Return the top of the stack
show()	Show the complete stack
pop()	Return the top value and remove from stack

[10]

(e) Based on the answer from (d), draw a diagram of the globalenv after the call **m <- mystack().**

[4]

2. (a) Show the general form for the magrittr operator for the function f(x,y). Describe the benefits of using the %>% operator.

[3]

(b) Describe each of the following dplyr functions: filter(), select(), summarise(), pull().

[4]

(c) Consider the following tables **x** and **y**.

Table x

```
    key val_x
  <dbl> <dbl>
1     1    10
2     2    20
3     3    30
4     5    40
```

Table y

```
    key val_y
  <dbl> <dbl>
1     1     5
2     2    10
3     6    15
4     7    20
```

Identify the relevant **dplyr** calls that generate the following tables.

(1)

```
    key val_y
  <dbl> <dbl>
1     1     5
2     2    10
```

(2)

```
  key val_x
  <dbl> <dbl>
1     3    30
2     5    40
```

(3)

```
    key val_y val_x
  <dbl> <dbl> <dbl>
1     1     5    10
2     2    10    20
3     6    15    NA
4     7    20    NA
```

(4)

```
    key val_x val_y
  <dbl> <dbl> <dbl>
1     1    10     5
2     2    20    10
```

[8]

(d) Consider the following tibble (called data), and the grading rules.

```
  Ids     CS101 CS102 CS103 CS104 CS105
  <chr>   <dbl> <dbl> <dbl> <dbl> <dbl>
1 1111111    80    50    51    70    61
2 1111112    70    60    71    81    56
3 1111113    24    54    48    58    68
4 1111114    55    91    49    39    69
5 1111115    62    46    59    45    76
```

< 40	Fail
>= 40 & < 50	Pass
>=50 & < 60	H2.2
>=60 & < 70	H2.1
>=70 & < 100	H1

Using dplyr and tidyr functions, create the following result.

```
  SubjectCode MeanResult    H1  H2.1  H2.2  Pass  Fail
  <chr>            <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
1 CS101             58.2    40    20    20     0    20
2 CS102             60.2    20    20    40    20     0
3 CS103             55.6    20     0    40    40     0
4 CS104             58.6    40     0    20    20    20
5 CS105             66      20    60    20     0     0
```

[10]

(3) (a) Describe the key differences between the S3 class system and message-passing OO systems such as Java and C++.

[4]

(b) Explain the following R function.

```
myarray <- function (n){
  structure(list(data=vector(mode = "numeric", length = n)),
            class="myarray")
}
```

[4]

(c) Use the function described in (b) to implement an S3 object system for a vector class. The following methods should be implemented for the new class.

Operator/Method	**Description**
[	Filter values from the array
[<-	Assign values to the array
summary	Summarise the array (see below)
<, <=, >, >=, ==, !=	Logical operators

As a guide, consider the following output.

```
> m <- myarray(10)
>
> m
$data
 [1] 0 0 0 0 0 0 0 0 0 0

attr(,"class")
[1] "myarray"


> summary(m)
Array Size =  10
Values = [ 0 0 0 0 0 0 0 0 0 0 ]

> m[1:4]<-7:10
> m[1:10]
 [1]  7  8  9 10  0  0  0  0  0  0
```

[12]

(d) Show how a new class can be created that inherits from a data frame. Show how a custom-built print function could be used to print the number of rows and columns of the new object, as well as any data.

[5]

(4) (a) Describe the different data types in base R, and show how they can be classified as either heterogeneous or homogenous.

[3]

(b) Predict the data types of the following vectors, and explain your answers.

```
c(10, 20, TRUE, "TRUE")

c(T,T,F,0)

unlist(list(10, 20, TRUE, "TRUE"))
```

[3]

(c) Describe the workings of this function, and explain how each line of code contributes to the output.

```
my_func <- function(x, f, ...) {
  out <- vector(mode = "list", length = length(x))
  for (i in seq_along(x)) {
    out[[i]] <- f(x[[i]], ...)
  }
  out
}
```

[6]

(d) Consider the following list:

```
l <- list(el1=1:3, el2="Test", el3=list(n1=10, n2=2:5))
```

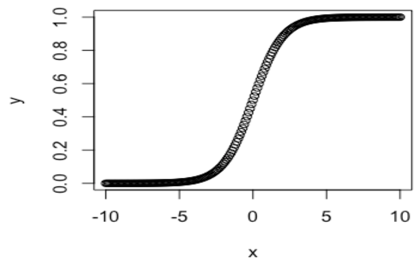
Draw a representation of the following commands, and explain each solution.

```
l[3]
l[1:2]
l[[1]]
l[[3]][[2]][3]
```

[8]

(e) Use sapply() to generate a sigmoid response to an input x [-10,10] in steps of 0.1

$$S(t) = \frac{1}{1 + e^{-x}}$$



[5]