

Examination Advanced R Programming

Linköpings Universitet, IDA, Statistik

Course code and name:	732A94 Advanced R Programming
Date:	2022/03/02, 8–12
Teacher:	Krzysztof Bartoszek phone 013–281 885
Allowed aids:	The extra material is included in the zip file exam_help_material_732A94.zip
Grades:	A= [18 – 20] points
	B= [16 – 18) points
	C= [14 – 16) points
	D= [12 – 14) points
	E= [10 – 12) points
	F= [0 – 10) points
Instructions:	Write your answers in an R script file named [your exam account].R The R code should be complete and readable code, possible to run by copying directly into a script. Comment directly in the code whenever something needs to be explained or discussed. Follow the instructions carefully. There are THREE problems (with sub-questions) to solve.

Problem 1 (5p)

a) (3p) The following function might run correctly, and return the square of its argument, but there are problems with it.

```
f_x2<-function(x){  
  res<-x*x  
  x<-paste0("The square of ",x,"is ", res,".\n")  
  cat(x)  
  x<-res  
  x  
}
```

Discuss what are the problems with this implementation.

b) (2p) Discuss proper ways of naming of variables and functions in one's code.

Problem 2 (10p)

READ THE WHOLE QUESTION BEFORE STARTING TO IMPLEMENT! Remember that your functions should **ALWAYS** check for correctness of user input!

a) (2p) In this task you should use object oriented programming in S3 or RC to write code that keeps track of people entering a swimming pool. It should contain information concerning the person, and the amount of people in the pool.

Your goal is to first initialize the swimming pool object, which should be of class "pool". Depending on your chosen OO system you can do it through a constructor (RC) or by implementing a function `create_pool()` (S3). The constructing function should take two integer arguments, `max_number_people`, the maximum number of people that can be in the pool at a time, and `number_presents` (more on this later). The pool object should contain for each entering person information, in particular a unique id of the person, a text name (you may have the same one e.g. "a" for every person), and their age. If needed, you may assume that the number of distinct people entering is bounded by some constant, and if needed set this constant yourself but it has to be greater than 20. Provide some example calls to your code.

```
## example call to create a pool object
my_pool <- create_workshop(max_number_people=100,number_presents=100) # S3
my_pool <- pool$new(max_number_people=100,number_presents=100) # RC
```

b) (3p) Now implement a function called `person_entering_pool()` that is invoked when a person arrives at the pool. The function should have two parameters: the person's name, and age. The id is to be automatically generated. Do not forget that the pool has a maximum capacity! Your code should react appropriately if the pool is already full. If a person is under the age of 16, then they should get a present. Remember that there is a finite number of presents (this is what the variable `number_presents` is for). Your code should react appropriately if the presents have run out. Provide some example calls to your code.

```
my_pool <- person_entering_pool(my_pool,"Bob",15) ## S3 and RC example call
## if using RC you may also call in this way
my_pool$person_entering_pool("Bob",15)
```

c) (3p) Now implement a function called `person_leaving_pool()` that is invoked when a person leaves the pool. The function should take one variable, the unique id of the leaving person. Remember to update the status of the pool when a person leaves. Provide some example calls to your code.

```
my_pool <- person_leaving_pool(my_pool,"id_1") ## S3 and RC example call
## if using RC you may also call in this way
my_pool$person_leaving_pool("id_1")
```

d) (2p) Implement a function that displays the state of the pool. You are free to choose yourself how to report the state! This function has to also work directly with `print()`.

```
# calls to show state of the pool
my_pool; print(my_pool)
```

Problem 3 (5p)

a) (2p) Consider the following sum,

$$f(N) = \sum_{n=1}^N \frac{1}{n^2 + n}.$$

Implement a function that takes as its input a positive integer number and returns the value of $f(N)$. The function should obtain the value of $f(N)$ by implementing the sum above, **NOT** using the closed form formula that will be provided in c). Provide example calls and results of runs of your code. Do not forget that your function should check for correctness of input and react appropriately. In the Computational Statistics course you were/will be told that the summation should be done in reverse, i.e. $n = N \dots 1$.

b) (1p) What is the computational complexity in terms of N of your code?

c) (2p) It is known also that

$$f(N) = 1 - \frac{1}{N+1}$$

Implement a unit test that compares your implementation of the summation with the above closed form formula for $f(N)$.