Review of Computer Literacy

1. final exam on Friday 4 August 2023, at 10AM
   - general information
   - policies for the final exam
2. some example questions
   - computer literacy terms
   - number systems and truth tables
   - algorithm design
   - modular design
   - object oriented design

MCS 260 Lecture 43
Introduction to Computer Science
Jan Verschelde, 2 August 2023

review of computer literacy

1. final exam on Friday 4 August 2023, at 10AM
   - general information
   - policies for the final exam

2. some example questions
   - computer literacy terms
   - number systems and truth tables
   - algorithm design
   - modular design
   - object oriented design

general information

The exam will take place on Friday 4 August, from 10AM till noon.

If an emergency prevents you from participation,
please contact me as soon as you are able to
so we can schedule a makeup exam in finals week.

The final exam is comprehensive and covers the entire course.

Please review the posted answers to the midterm and the quizzes.

review of computer literacy

1. final exam on Friday 4 August 2023, at 10AM
   - general information
   - policies for the final exam

2. some example questions
   - computer literacy terms
   - number systems and truth tables
   - algorithm design
   - modular design
   - object oriented design

policies for the final exam

The exam will be online, similar to the midterm exam.

1. The questions will be emailed to you.
2. Your answers must be in a Jupyter notebook, uploaded into gradescope.

The material breaks down in two parts:

1. computer science concepts: define a term, give an example, explain the difference.
2. mathematical CS: binary and hexadecimal representations of numbers, boolean algebra, truth tables, design of algorithms and programs via flowcharts, principles of good modular design, UML diagrams.

This review contains some preliminary examples of questions which may help you prepare for the literacy part of the final exam.

review of computer literacy

1. final exam on Friday 4 August 2023, at 10AM
   - general information
   - policies for the final exam

2. some example questions
   - computer literacy terms
   - number systems and truth tables
   - algorithm design
   - modular design
   - object oriented design

computer literacy terms

See the posted script `literacy.py`.

This script contains a dictionary of a bit more than 200 terms, organized according to each lecture.

The main function in the script pulls a term at random and as hint shows the lecture number where the term was explained.

Review: run the script and consult the lecture slides if needed.

run the script `literacy.py` ...

... and see if you can explain the terms *in your own words.*

Running the script provides inspiration for questions:

- `hash function`: what does hashing mean?
- `GPL`: difference with public domain?
- `linker`: its relation with a compiler?
- `segmentation`: is it the same as pagination?
- `complexity class`: can your problem be in a bad class?
- `fork a process`: what is the analogue for a thread?
- `beta testing`: is this whitebox or blackbox, or both?
- `portable code`: why is it important?
- `language generation`: how many generations do we know?
- `socket`: what is a socket good for?

review of computer literacy

1. final exam on Friday 4 August 2023, at 10AM
   - general information
   - policies for the final exam

2. some example questions
   - computer literacy terms
   - number systems and truth tables
   - algorithm design
   - modular design
   - object oriented design

number systems and truth tables

- The binary and hexadecimal number system.
  1. Convert 118 into binary notation.
  2. Convert 2890 into hexadecimal notation.

  For both problems, show all *computational* steps.

- Logic expressions and truth tables.

  A truth table justifies the simplification of logic expressions.
  1. For n variables, the truth table has 2^n rows.
  2. There are as many columns as operands in the expressions.

truth tables

Use truth tables to verify that

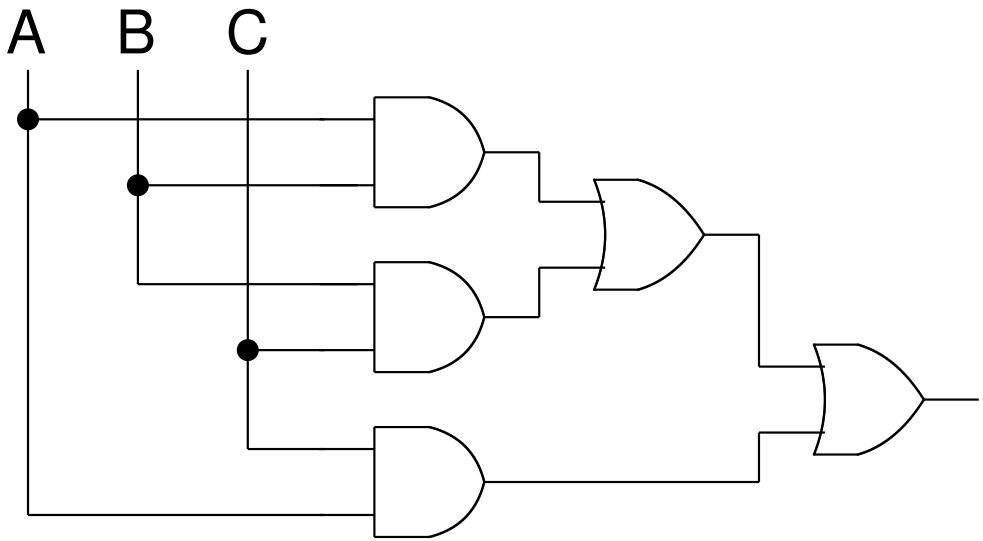
$$((\text{NOT } A) \text{ AND } B \text{ AND } C) \text{ OR } (A \text{ AND } (\text{NOT } B) \text{ AND } C) \\ \text{OR } (A \text{ AND } B \text{ AND } (\text{NOT } C)) \text{ OR } (A \text{ AND } B \text{ AND } C)$$

is equivalent to

$$(A \text{ AND } B) \text{ OR } (B \text{ AND } C) \text{ OR } (A \text{ AND } C).$$

Draw a realization of the second expression
using the diagrams for the logic gates.

consider the circuit



Take 0 for A, the value 1 for B, and 0 for C.
What does the circuit compute?
Write the logical expression that corresponds with this circuit.

review of computer literacy

1. final exam on Friday 4 August 2023, at 10AM
   - general information
   - policies for the final exam

2. some example questions
   - computer literacy terms
   - number systems and truth tables
   - algorithm design
   - modular design
   - object oriented design

algorithm design

Consider the following pseudocode.

```
Input: a list c of numbers, len(c) > 0, and a number x.
Output: the sum of c[k]*x**k for k in range(len(c)).

n = len(c)
s = c[n-1]
for k in the range n-2, n-3, .., 0 do
    s = s*x + c[k]
```

1. Draw the flowchart for the algorithm.
2. What is the cost of this algorithm?

review of computer literacy

1. final exam on Friday 4 August 2023, at 10AM
   - general information
   - policies for the final exam

2. some example questions
   - computer literacy terms
   - number systems and truth tables
   - algorithm design
   - modular design
   - object oriented design

design the modular structure

Consider the modular design of a program for an online quiz on Calculus I and Calculus II. The design could be done in two different ways, described below.

1. There are two modules, called `Calculus I` and `Calculus II`. The module `Calculus I` stores questions and answers for topics covered in Calculus I. Questions and answers for Calculus II are in the other module, called `Calculus II`.
2. There are two modules, called `Questions` and `Answers`. The module `Questions` stores all questions for the topics covered in Calculus I and Calculus II. Answers to all questions are defined by the module `Answers`.

Which design is best? 1 or 2?
Justify your choice by referring to the key principles of modular design.

review of computer literacy

1. final exam on Friday 4 August 2023, at 10AM
   - general information
   - policies for the final exam

2. some example questions
   - computer literacy terms
   - number systems and truth tables
   - algorithm design
   - modular design
   - object oriented design

object oriented design

Describe the object oriented design of a car rental company.

After logging in, employees can add or delete customers, add or delete cars, in addition to all operations also all customers are allowed to do.

Customers can view the cars that are currently available,
they can check out or return a car.

Use UML to define

- class diagrams and
- use case diagrams

to model the car rental company.