NYU, Tandon School of Engineering
Extended Bridge to CS — Winter 2020

Exam #1

Thursday, February 13, 2020

- This exam has 12 questions, with 100 points total.
- You have **two hours**.
- **You should submit your answers to the corresponding places in the exam on the NYU Classes system.**
- **It is your responsibility to take the time for the exam** (You may use a timer).
  After the time is up, the exam will be automatically disabled for submissions.
  Make sure to upload the files with your answers BEFORE the time is up (the files will NOT be uploaded automatically).
  We will not accept any late submissions.
- In total, you should upload 3 '.cpp' files:
  - One '.cpp' file for questions 1-10.
    Write your answer as one long comment (/* … */).
    Name this file 'YourNetID_q1to10.cpp'.
  - One '.cpp' file for question 11, containing your code.
    Name this file 'YourNetID_q11.cpp'.
  - One '.cpp' file for question 12, containing your code.
    Name this file 'YourNetID_q12.cpp'.
- **Write your name, and netID at the head of each file.**
- This is a closed-book exam. However, you are allowed to use
  CLion or Visual-Studio. You should create a new project, and work ONLY in it.
  You may also use two sheets of scratch paper.
  Besides that, no additional resources (of any form) are allowed.
- Calculators are **not** allowed.
- Read every question completely before answering it.
  Note that there are 2 programming problems at the end**.**
  Be sure to allow enough time for these questions

Table 1.5.1: Laws of propositional logic.

Idempotent laws:	$p \vee p \equiv p$	$p \wedge p \equiv p$
Associative laws:	$(p \vee q) \vee r \equiv p \vee (q \vee r)$	$(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$
Commutative laws:	$p \vee q \equiv q \vee p$	$p \wedge q \equiv q \wedge p$
Distributive laws:	$p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$	$p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$
Identity laws:	$p \vee F \equiv p$	$p \wedge T \equiv p$
Domination laws:	$p \wedge F \equiv F$	$p \vee T \equiv T$
Double negation law:	$\neg\neg p \equiv p$	
Complement laws:	$p \wedge \neg p \equiv F$ $\neg T \equiv F$	$p \vee \neg p \equiv T$ $\neg F \equiv T$
De Morgan's laws:	$\neg(p \vee q) \equiv \neg p \wedge \neg q$	$\neg(p \wedge q) \equiv \neg p \vee \neg q$
Absorption laws:	$p \vee (p \wedge q) \equiv p$	$p \wedge (p \vee q) \equiv p$
Conditional identities:	$p \rightarrow q \equiv \neg p \vee q$	$p \leftrightarrow q \equiv (p \rightarrow q) \wedge (q \rightarrow p)$

Table 1.12.1: Rules of inference known to be valid arguments.

Rule of inference	Name
p $p \rightarrow q$ $\therefore q$	Modus ponens
$\neg q$ $p \rightarrow q$ $\therefore \neg p$	Modus tollens
p $\therefore p \vee q$	Addition
$p \wedge q$ $\therefore p$	Simplification

Rule of inference	Name
p q $\therefore p \wedge q$	Conjunction
$p \rightarrow q$ $q \rightarrow r$ $\therefore p \rightarrow r$	Hypothetical syllogism
$p \vee q$ $\neg p$ $\therefore q$	Disjunctive syllogism
$p \vee q$ $\neg p \vee r$ $\therefore q \vee r$	Resolution

Table 1.13.1: Rules of inference for quantified stateme

Rule of Inference	Name
c is an element (arbitrary or particular) $\underline{\forall x\ P(x)}$ $\therefore$ P(c)	Universal instantiation
c is an arbitrary element $\underline{P(c)\qquad}$ $\therefore \forall x\ P(x)$	Universal generalization
$\underline{\exists x\ P(x)}$ $\therefore$ (c is a particular element) $\wedge$ P(c)	Existential instantiation*
c is an element (arbitrary or particular) $\underline{P(c)\qquad}$ $\therefore \exists x\ P(x)$	Existential generalization

Table 3.6.1: Set identities.

Name	Identities	
Idempotent laws	$A \cup A = A$	$A \cap A = A$
Associative laws	$(A \cup B) \cup C = A \cup (B \cup C)$	$(A \cap B) \cap C = A \cap (B \cap C)$
Commutative laws	$A \cup B = B \cup A$	$A \cap B = B \cap A$
Distributive laws	$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$	$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$
Identity laws	$A \cup \varnothing = A$	$A \cap U = A$
Domination laws	$A \cap \varnothing = \varnothing$	$A \cup U = U$
Double Complement law	$\overline{\overline{A}} = A$	
Complement laws	$A \cap \overline{A} = \varnothing$ $\overline{U} = \varnothing$	$A \cup \overline{A} = U$ $\overline{\varnothing} = U$
De Morgan's laws	$\overline{A \cup B} = \overline{A} \cap \overline{B}$	$\overline{A \cap B} = \overline{A} \cup \overline{B}$
Absorption laws	$A \cup (A \cap B) = A$	$A \cap (A \cup B) = A$

Part I – Theoretical:

- ***You don't need to justify your answers to the questions in this part.***
- *For all questions in this part of the exam (questions 1-10), you should submit a **single** '.cpp' file. Write your answers as one long comment (`/* … */`).*
  Name this file 'YourNetID_q1to10.cpp'.

Question 1 (5 points)

a. Convert the decimal number $(178)_{10}$ to its base-2 representation.
b. Convert the 8-bits two's complement number $(11101100)_{\text{8-bit two's complement}}$ to its decimal representation.

Question 2 (5 points)

Select the statement that is equivalent to:
"It is not true that the patient doesn't have high blood pressure and doesn't have influenza."
a. The patient has high blood pressure and has influenza.
b. The patient doesn't have high blood pressure and doesn't have influenza.
c. The patient doesn't have high blood pressure or doesn't have influenza.
d. The patient has high blood pressure or has influenza.

Question 3 (5 points)

The domain for variable x is the set {Ann, Ben, Cam, Dave}.
The table below gives the values of predicates P and Q for every element in the domain.

	P(x)	Q(x)
Ann	F	F
Ben	F	T
Cam	T	F
Dave	F	T

Select the statement that is **true**.
a. $\forall x\ (Q(x) \rightarrow P(x))$
b. $\forall x\ (P(x) \wedge Q(x))$
c. $\exists x\ (P(x) \rightarrow Q(x))$
d. $\exists x\ (P(x) \wedge Q(x))$

Question 4 (5 points)

The domain of discourse for x and y is the set of employees at a company. *Miguel* is one of the employees at the company.

Define the predicates:

V(x): x is a manager

M(x, y): x earns more than y

Select the logical expression that is equivalent to:

"Everyone who earns more than Miguel is a manager."

a. $\forall x\ (M(x, Miguel)\ \rightarrow\ V(x))$

b. $\forall x\ (M(x, Miguel)\ \wedge\ \neg V(x))$

c. $\neg\exists x\ (M(V(x), Miguel)$

d. $\neg\exists x\ (M(x, Miguel)\ \wedge\ V(x))$

Question 5 (5 points)

The domain of discourse are the students in a class. Define the predicates:

S(x): x studied for the test

A(x): x received an A on the test

Select the logical expression that is equivalent to:

"Someone who did not study for the test received an A on the test."

a. $\exists x\ (A(x) \rightarrow \neg S(x))$

b. $\exists x\ (\neg S(x) \rightarrow A(x))$

c. $\exists x\ (\neg S(x) \wedge A(x))$

d. $\exists x\ (\neg S(x) \leftrightarrow A(x))$

Question 6 (5 points)

Select the logical expression that is equivalent to: $\neg\forall x\ \exists y\ (P(x) \wedge Q(x, y))$

a. $\exists y\ \forall x\ (\neg P(x) \vee \neg Q(x, y))$

b. $\forall y\ \exists x\ (\neg P(x) \vee \neg Q(x, y))$

c. $\exists x\ \forall y\ (\neg P(x) \vee \neg Q(x, y))$

d. $\forall x\ \exists y\ (\neg P(x) \vee \neg Q(x, y))$

Question 7 (5 points)

Let A = {1, 2, 3, 4}. Select the statement that is **false**.

a. $\emptyset \in P(A)$
b. $\emptyset \subseteq P(A)$
c. $\{2,3\} \in P(A)$
d. $\{2,3\} \subseteq P(A)$

Question 8 (10 points)

$A = \left\{1, \{2\}, \left\{\{3, 4\}\right\}\right\}$.

For each of the following statements, state if they are **true or false** (no need to explain your choice).

a. $1 \in A$
b. $1 \subseteq A$
c. $\{2\} \in A$
d. $\{2\} \subseteq A$
e. $\{3, 4\} \in A$
f. $\{3, 4\} \subseteq A$
g. $\left\{\{3, 4\}\right\} \in A$
h. $\left\{\{3, 4\}\right\} \subseteq A$
i. $\emptyset \in A$
j. $\emptyset \subseteq A$

Question 9 (5 points)

Consider the following function:
$f: \{0,1\}^4 \to \{0,1\}^3$. f(x) is obtained from x by removing the first bit.
For example, f(1010) = 010.
Select the correct description of the function f.
a. One-to-one and onto
b. One-to-one but not onto
c. Onto but not one-to-one
d. Neither one-to-one nor onto

Question 10 (5 points)

Let A={1, 2, 3}.
The function $f: P(A) \to P(A)$ is defined as: $f(X) = X - \{1\}$.
For example, $f(\{1, 2\}) = \{2\}$ (Since: {1, 2} – {1} is {2}),

Find the range of f

Part II – Coding:

- *For **each** question in this part (questions 11-12), you should submit a '.cpp' file, containing your code.*
- *Pay special attention to the style of your code. Indent your code correctly, choose meaningful names for your variables, define constants where needed, choose most suitable control statements, etc.*
- *In all questions, you may assume that the user enters inputs as they are asked.*
  For example, if the program expects a positive integer, you may assume that user will enter positive integers.
- *No need to document your code. However, you may add comments if you think they are needed for clarity.*

Question 11 (20 points)

Write a program that reads an odd integer greater or equal to 3, n, and prints a shape of a *n*-line hollow square made of '#' symbols, with a '$' symbol shape of a X, bounded in it.

Your program should interact with the user **exactly** as demonstrated in the following two executions:

Execution example 1:
```
Please enter an odd integer, greater or equal to 3:
7
#######
#$   $#
# $ $ #
#  $  #
# $ $ #
#$   $#
#######
```

Execution example 2:
```
Please enter an odd integer, greater or equal to 3:
5
#####
#$ $#
# $ #
#$ $#
#####
```

<u>Question 12 (25 points)</u>

In this question, you will write a program that reads from the user grades of some students and reports the average of each student.

Your program should first ask the user how many students they have in the class. For each student, the user should be asked to enter a line containing a sequence of the student's grades. The grades should be separated by a space, and end with -1 (to indicate the end of the grades list). After reading the grades, your program should print the average of that student's grades.

Your program should interact with the user **exactly** the same way, as demonstrated bellow (Specifically, make sure that when announcing the average, your program will generate a sentence including the serial number of that student):

```
How many students do you wish to enter?
4

Enter list of grades, separated by a space. End the grades list by typing -1:
71  98  89  -1
Average of student #1 is 86.0

Enter list of grades, separated by a space. End the grades list by typing -1:
92  89  95  86  -1
Average of student #2 is 90.5

Enter list of grades, separated by a space. End the grades list by typing -1:
85  92  100  94  87  -1
Average of student #3 is 91.6

Enter list of grades, separated by a space. End the grades list by typing -1:
93  67  -1
Average of student #4 is 80.0
```