THE UNIVERSITY OF BRITISH COLUMBIA
CPSC 203: MIDTERM EXAMINATION – October 24, 2019

Full Name: _____ CS Account: _____

Signature: _____ UBC Student #: |_|_|_|_|_|_|_|_|_|

Important notes about this examination

1. You have 80 minutes to complete this exam.
2. No notes or electronics of any kind are allowed.
3. Good luck!

Student Conduct during Examinations

1. Each examination candidate must be prepared to produce, upon the request of the invigilator or examiner, his or her UBCcard for identification.
2. Examination candidates are not permitted to ask questions of the examiners or invigilators, except in cases of supposed errors or ambiguities in examination questions, illegible or missing material, or the like.
3. No examination candidate shall be permitted to enter the examination room after the expiration of one-half hour from the scheduled starting time, or to leave during the first half hour of the examination. Should the examination run forty-five (45) minutes or less, no examination candidate shall be permitted to enter the examination room once the examination has begun.
4. Examination candidates must conduct themselves honestly and in accordance with established rules for a given examination, which will be articulated by the examiner or invigilator prior to the examination commencing. Should dishonest behaviour be observed by the examiner(s) or invigilator(s), pleas of accident or forgetfulness shall not be received.
5. Examination candidates suspected of any of the following, or any other similar practices, may be immediately dismissed from the examination by the examiner/invigilator, and may be subject to disciplinary action:
   i. speaking or communicating with other examination candidates, unless otherwise authorized;
   ii. purposely exposing written papers to the view of other examination candidates or imaging devices;
   iii. purposely viewing the written papers of other examination candidates;
   iv. using or having visible at the place of writing any books, papers or other memory aid devices other than those authorized by the examiner(s); and,
   v. using or operating electronic devices including but not limited to telephones, calculators, computers, or similar devices other than those authorized by the examiner(s)—(electronic devices other than those authorized by the examiner(s) must be completely powered down if present at the place of writing).
6. Examination candidates must not destroy or damage any examination material, must hand in all examination papers, and must not take any examination material from the examination room without permission of the examiner or invigilator.
7. Notwithstanding the above, for any mode of examination that does not fall into the traditional, paper-based method, examination candidates shall adhere to any special rules for conduct as established and articulated by the examiner.
8. Examination candidates must follow any additional examination rules or directions communicated by the examiner(s) or invigilator(s).

Please do not write in this space:

CPSC 203 2019W1: Midterm Exam

October 24, 2019

1 Who gets the marks? [1 mark] Write your 4 or 5 digit CSID here

2 All Work and No Play List [15-marks]

Suppose you are writing a Python program whose purpose is to organize your music library. Use the template below to design the class definitions you would use to structure your data. We have given you more than enough entries — some of the blanks may remain empty! Notes: a) The marks for this problem will depend on design, not syntax. b) If a phrase is not familiar to you, make a guess and explain your ideas. c) We are only interested in the declarations for the member functions, not their implementations.

1. **[9-marks]** Each of the phrases below represents some component of your program's design. Some are class names, some are member variable names, and some are member function names. Add the phrases to the class definition template. For any member function, please give simple names for reasonable parameters, if needed.

```
title     playList  songs
favorite  play      artist
play      song      removeSong
```

2. **[6-marks]** Now add as much type information as you can to each of the elements of the class definition.

```
@dataclass
class _________________:
        _________________:_________________
        _________________:_________________
        _________________:_________________
        def _____________________________________:
        def _____________________________________:
        def _____________________________________:

@dataclass
class _________________:
        _________________:_________________
        _________________:_________________
        _________________:_________________
        def _____________________________________:
        def _____________________________________:
        def _____________________________________:
```

3 Free Lunch [6-marks]

Please answer these questions as thoughtfully as you can, given the circumstances. You are guaranteed to receive at least 4 marks.

1. **[2-marks]** Suppose you have forgotten the name of the Beautiful Soup library. What google query would you use to find it again?

2. **[2-marks]** Using two sentences, describe the biggest challenge you faced in completing Project 1: Billboard.

3. **[2-marks]** On a scale of 1 to 5, where 5 is the most, how much did you learn in Project 1: Billboard?

   ◯ 1 ◯ 2 ◯ 3 ◯ 4 ◯ 5

4 Picture Frames [8-marks]

This problem depends on the file `art.csv` file (shown on the right) which we read into a pandas dataframe using these Python statements:

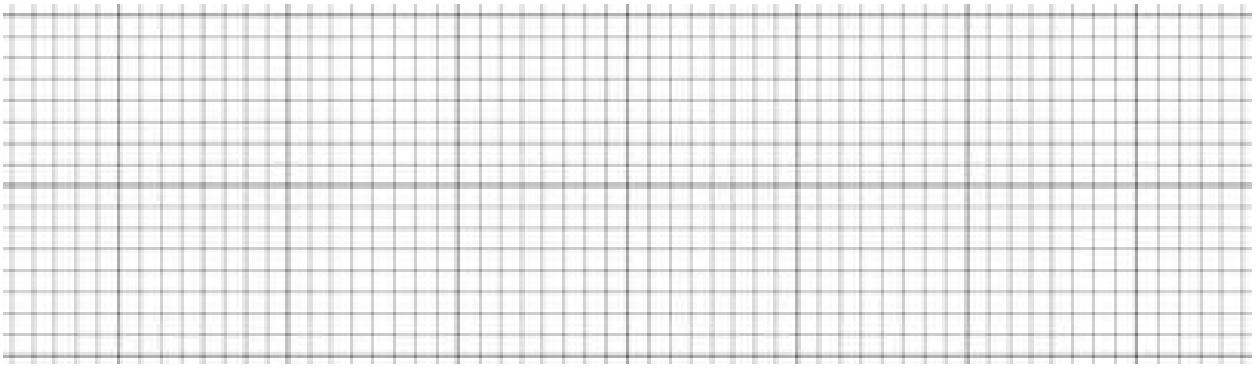
```
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv('art.csv')
```

Art ID	Medium	Year	Sales	Subject	Cost
A001	Oil	2000	123	Pastoral	350
A002	Water	2006	114	Portrait	450
A003	Water	2003	135	Abstract	169
A004	Oil	1996	139	Still Life	189
A005	Water	2010	117	Still Life	183
A006	Oil	2002	121	Pastoral	80
A007	Oil	1998	133	Abstract	166
A008	Water	1992	140	Pastoral	120
A009	Oil	1998	133	Pastoral	75
A010	Oil	2002	133	Still Life	40

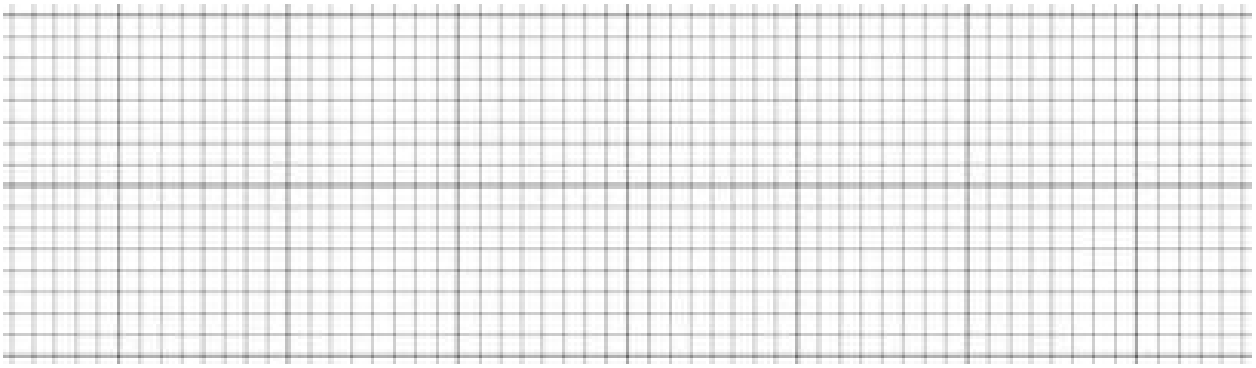
1. **[4-marks]** In the space below, sketch the result of the following code. (Your sketch need not be perfect!)

```
df.plot(kind='scatter', x='Year', y='Sales')
plt.show()
```



2. **[4-marks]** In the space below, sketch the result of the following code. (Your sketch need not be perfect!)

```
v1 = df.groupby(['Subject', 'Medium']).Sales.sum()
v1.unstack().plot(kind='bar', stacked=True,  color=['red','blue'])
plt.show()
```



5 You've Been Framed [10-marks]

This problem depends on the file `art.csv` file (shown on the right) which we read into a pandas dataframe using these Python statements:

```
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv('art.csv')
df.astype({
    'Art ID': 'str', 'Medium': 'str',
    'Year': 'int64', 'Sales': 'int64',
    'Subject': 'str', 'Cost': 'int64'
    })
```

Art ID	Medium	Year	Sales	Subject	Cost
A001	Oil	2000	123	Pastoral	350
A002	Water	2006	114	Portrait	450
A003	Water	2003	135	Abstract	169
A004	Oil	1996	139	Still Life	189
A005	Water	2010	117	Still Life	183
A006	Oil	2002	121	Pastoral	80
A007	Oil	1998	133	Abstract	166
A008	Water	1992	140	Pastoral	120
A009	Oil	1998	133	Pastoral	75
A010	Oil	2002	133	Still Life	40

Use `df` to answer the questions below.

1. **[2-marks]** Write a single line of Python code that creates a dataframe consisting of all the rows from `df` whose `Year` is greater than 2000.

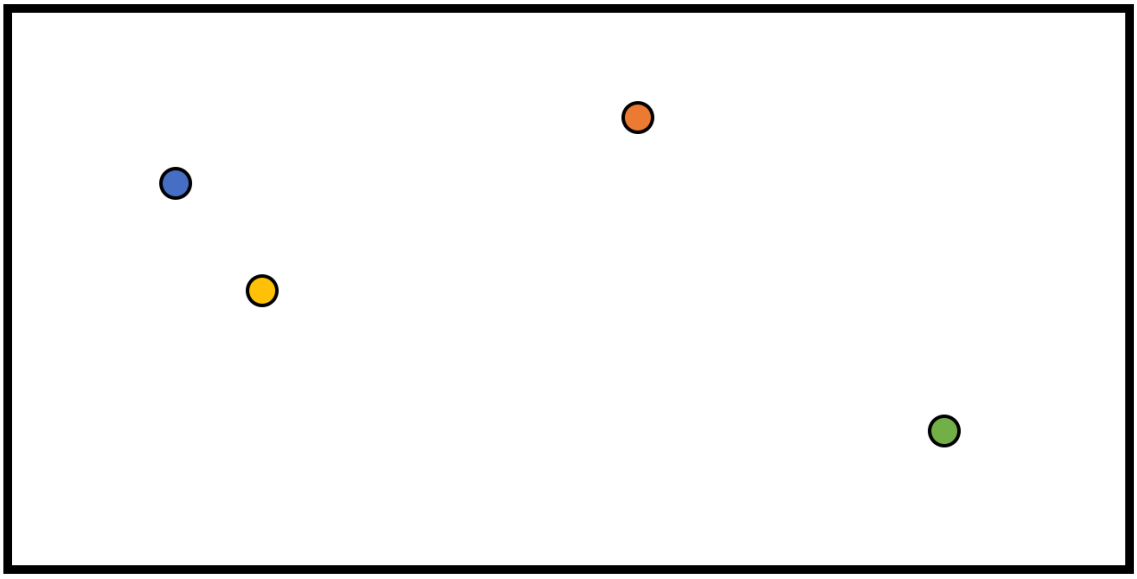
2. **[2-marks]** Write a single line of Python code that returns the *number* of rows in the dataframe for which the `Subject` is `Abstract`.

3. **[3-marks]** Write a single line of Python code that returns the average `Cost` of the rows in the dataframe for which the `Medium` is `Water`.

4. **[3-marks]** Write a single line of Python code that returns the `Subject` of the row in the dataframe whose `Cost` is largest.

6 Voronoi [8-marks]

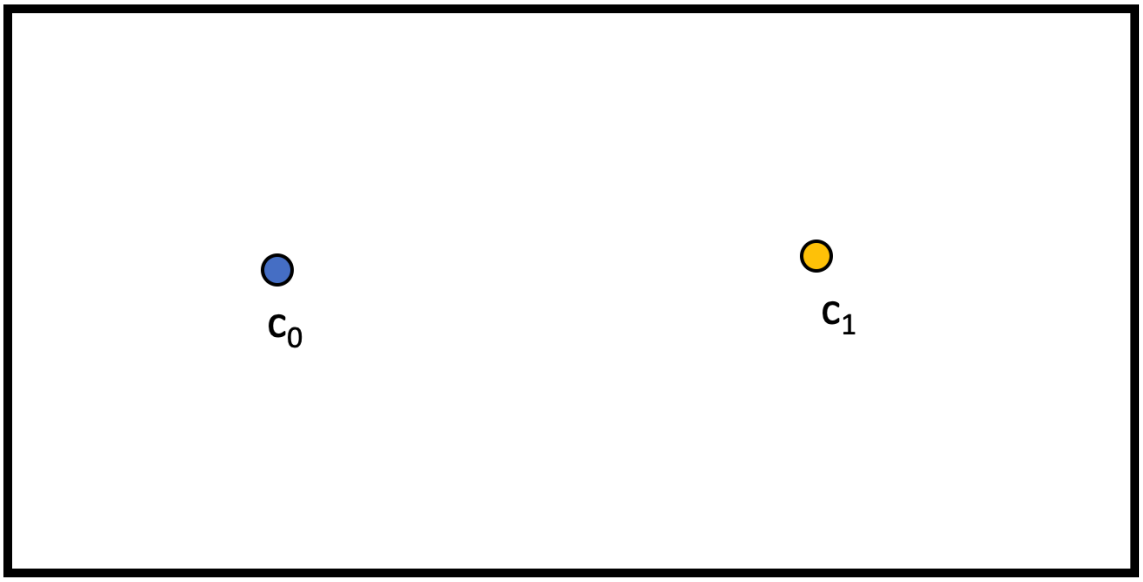
1. **[4-marks]** Sketch the Voronoi Diagram of this image as accurately as you can.



2. **[4-marks]** In this problem we will define a new way of partitioning a planar region (like an image). Instead of using a single distance formula, as we did for Voronoi, we will use a different formula for every center! Since our example has two centers, we will use two formulas, described in this table:

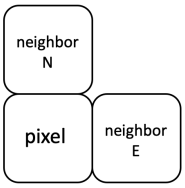
Center	Distance Formula	Description
$c_0 = (x_0, y_0)$	$d_0(x, y) = \sqrt{(x - x_0)^2 + (y - y_0)^2}$	The usual Pythagorean-based distance.
$c_1 = (x_1, y_1)$	$d_1(x, y) = \sqrt{(x - x_1)^2 + (y - y_1)^2}/2$	The usual, divided by 2.

Partition the region below so that each point is contained in the region corresponding to its nearest center, based on the given distance formulas.



7 Your Turn [12-marks]

In this problem we will explore a modification of the fill algorithm that we learned in class (to speed up the computation of a Voronoi Diagram). Namely, instead of considering the neighbors of a pixel to be the 4 adjacent pixels, each pixel has only 2 neighbors, the pixel above (N) and the pixel to the right (E):



1. **[2-marks]** What abstract structure will we use to organize the order in which we process the pixels?

2. **[2-marks]** Describe the location of the two neighbors of the pixel located at (x, y). Don't forget that the origin is in the upper left corner of the image!

3. **[4-marks]** Suppose you are coloring the image represented by this grid, starting from the one marked cell. We refer to this cell as a *center*, even though it's not in the center of the diagram. Lightly shade all of the cells colored by the algorithm (using the new definition of neighbors). Assume that neighbor N is processed before neighbor E.

00	10	20	30	40	50	60	70
01	11	21	31	41	51	61	71
02	12	22	32	42	52	62	72
03	13	23	33	43	53	63	73
04	14	24	34	44	54	64	74
05	15	25	35	45	55	65	75
06	16	26	36	46	56	66	76
07	17	27	37	47	57	67	77

4. **[4-marks]** Your answer to the previous part left some of the cells in the grid uncolored. How many additional centers do you need in order for the algorithm to completely fill the grid?

   Illustrate your answer by marking any necessary new center with an X in the grid above.

8 Making Connections [12-marks]

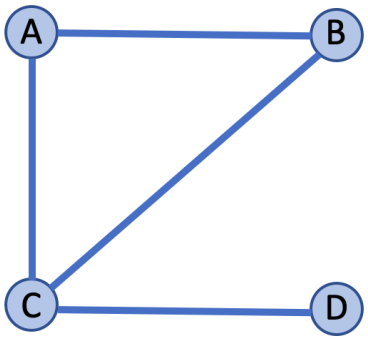
1. **[4-marks]**

   (a) Suppose a new friend tells you about their favorite graph that has 5 vertices, each of which has 3 neighbors (incident edges). Which of the following is true?
   - ◯ The new friend is lying. No such graph can exist.
   - ◯ The new friend is boring. There are lots of different graphs like that.
   - ◯ The new friend is dazzling. There is only one graph like that!

   (b) Suppose a friend tells you about their favorite graph that has 6 vertices and 15 edges.
   - ◯ The friend is lying. No such graph can exist.
   - ◯ The friend is boring. There are lots of different graphs like that.
   - ◯ The friend is dazzling. There is only one graph like that!

   (c) Suppose an old friend tells you about their favorite graph that has 53 vertices, each of which has 17 neighbors (incident edges).
   - ◯ The old friend is lying. No such graph can exist.
   - ◯ The old friend is boring. There are lots of different graphs like that.
   - ◯ The old friend is dazzling. There is only one graph like that!

2. **[8-marks]** In this problem we investigate some of the characteristics of a particular kind of graph. Part of the work you'll do requires the definition of a *path*. A path from vertex u to vertex v is a sequence of vertices and (non-repeating) edges along which you can travel to get from u to v. The first 3 questions below relate to this graph:



   (a) Give a list of vertices along a path from vertex B to vertex D: _____

   (b) A graph is said to be *connected* if there is a path between every pair of its vertices.
   Is the given graph connected? ◯ Yes ◯ No
   If you answer no, then circle two vertices between which there is no path.

   (c) A graph is said to have a *cycle* if there are two or more paths between a pair of vertices.
   Does the given graph part have a cycle? ◯ Yes ◯ No
   If you answer yes, then underline two vertices between which there are two different paths.

(d) Draw a connected graph with no cycles containing 5 vertices. How many edges does it have?

(e) Sketch 3 more connected graphs with no cycles containing 4, 5, and 6 vertices. In each case observe how many edges the graphs contain.

(f) Make a general statement about the number of edges in a connected graph with no cycles containing k vertices.

Connected graphs with no cycles are called *trees*.

This page intentionally left (almost) blank.
If you write answers here, you must CLEARLY indicate on this page what question they belong with AND on the problem's page that you have answers here.

This page intentionally left (almost) blank.
If you write answers here, you must CLEARLY indicate on this page what question they belong with AND on the problem's page that you have answers here.