Week7: First Tutorial

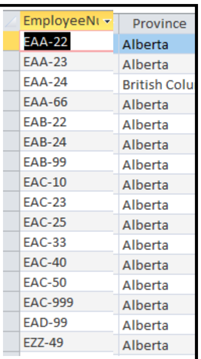
- Logic: student exercises (AND, OR, NOT)

Student Exercise #1

- Using the graphical design view of Access show the first name, last name and hourly pay rate of employees whose hourly earnings are in the range $25 - $50
- Form the equivalent query using SQL
- TA provides ~5 minutes to complete the exercise

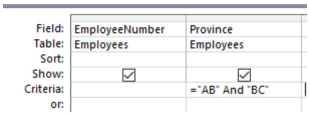
Author: James Tam

Student Exercise #2



EmployeeN	Province
EAA-22	Alberta
EAA-23	Alberta
EAA-24	British Colu
EAA-66	Alberta
EAB-22	Alberta
EAB-24	Alberta
EAB-99	Alberta
EAC-10	Alberta
EAC-23	Alberta
EAC-25	Alberta
EAC-33	Alberta
EAC-40	Alberta
EAC-50	Alberta
EAC-999	Alberta
EAD-99	Alberta
EZZ-49	Alberta

- What records will appear with the following query
- Graphical Design View



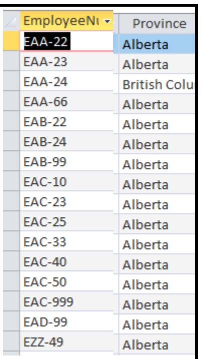
Field:	EmployeeNumber	Province
Table:	Employees	Employees
Sort:		
Show:	☑	☑
Criteria:		="AB" And "BC"
or:		

- SQL

```
SELECT Employees.EmployeeNumber, Employees.Province
FROM Employees
WHERE ((Employees.Province="AB") And
(Employees.Province="BC"));
```

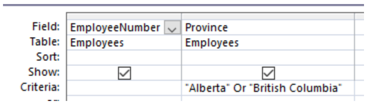
Author: James Tam

Student Exercise #3



EmployeeN	Province
EAA-22	Alberta
EAA-23	Alberta
EAA-24	British Colu
EAA-66	Alberta
EAB-22	Alberta
EAB-24	Alberta
EAB-99	Alberta
EAC-10	Alberta
EAC-23	Alberta
EAC-25	Alberta
EAC-33	Alberta
EAC-40	Alberta
EAC-50	Alberta
EAC-999	Alberta
EAD-99	Alberta
EZZ-49	Alberta

- What records will appear with the following query
- Graphical Design View



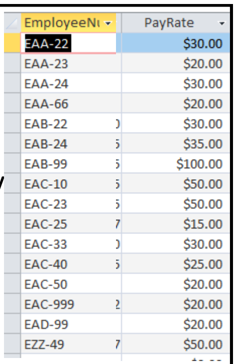
Field:	EmployeeNumber	Province
Table:	Employees	Employees
Sort:		
Show:	☑	☑
Criteria:		"Alberta" Or "British Columbia"

- SQL

```
SELECT Employees.EmployeeNumber, Employees.Province
FROM Employees
WHERE ((Employees.Province ="Alberta") OR
(Employees.Province="British Columbia"));
```

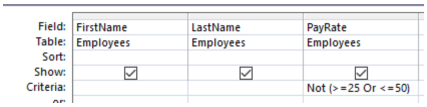
Author: James Tam

Student Exercise #4



EmployeeN	PayRate
EAA-22	$30.00
EAA-23	$20.00
EAA-24	$30.00
EAA-66	$20.00
EAB-22	$30.00
EAB-24	$35.00
EAB-99	$100.00
EAC-10	$50.00
EAC-23	$50.00
EAC-25	$15.00
EAC-33	$30.00
EAC-40	$25.00
EAC-50	$20.00
EAC-999	$20.00
EAD-99	$20.00
EZZ-49	$50.00

- What records will appear with the following query
- Graphical Design View



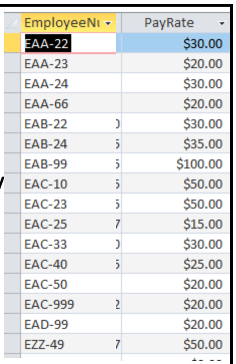
Field:	FirstName	LastName	PayRate
Table:	Employees	Employees	Employees
Sort:			
Show:	☑	☑	☑
Criteria:			Not (>=25 Or <=50)

- SQL

```
SELECT Employees.FirstName, Employees.LastName,
Employees.PayRate
FROM Employees
WHERE (Not ((Employees.PayRate >=25) Or
(Employees.PayRate<=50));
```

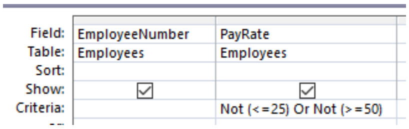
Author: James Tam

Student Exercise #5



EmployeeN	PayRate
EAA-22	$30.00
EAA-23	$20.00
EAA-24	$30.00
EAA-66	$20.00
EAB-22	$30.00
EAB-24	$35.00
EAB-99	$100.00
EAC-10	$50.00
EAC-23	$50.00
EAC-25	$15.00
EAC-33	$30.00
EAC-40	$25.00
EAC-50	$20.00
EAC-999	$20.00
EAD-99	$20.00
EZZ-49	$50.00

- What records will appear with the following query
- Graphical Design View



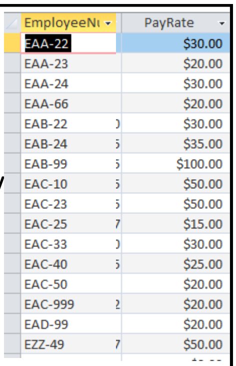
Field:	EmployeeNumber	PayRate
Table:	Employees	Employees
Sort:		
Show:	☑	☑
Criteria:		Not (<=25) Or Not (>=50)

- SQL

```
SELECT Employees.EmployeeNumber, Employees.PayRate
FROM Employees
WHERE ((Not (Employees.PayRate<=25) Or Not
(Employees.PayRate>=50)));
```

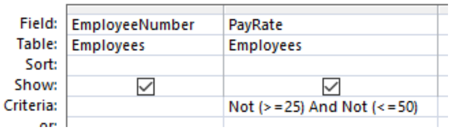
Author: James Tam

Student Exercise #6



EmployeeNu		PayRate
EAA-22		$30.00
EAA-23		$20.00
EAA-24		$30.00
EAA-66		$20.00
EAB-22	)	$30.00
EAB-24	5	$35.00
EAB-99	5	$100.00
EAC-10	5	$50.00
EAC-23	5	$50.00
EAC-25	7	$15.00
EAC-33	)	$30.00
EAC-40	5	$25.00
EAC-50		$20.00
EAC-999	2	$20.00
EAD-99		$20.00
EZZ-49	7	$50.00

- What records will appear with the following query
- Graphical Design View



Field:	EmployeeNumber	PayRate
Table:	Employees	Employees
Sort:		
Show:	☑	☑
Criteria:		Not (>=25) And Not (<=50)

- SQL

```
SELECT Employees.EmployeeNumber, Employees.PayRate
FROM Employees
WHERE ((Not (Employees.PayRate>=25) And Not
(Employees.PayRate<=50)))
```

Author: James Tam

Student Exercise #7A: Query With Multiple Criteria

- Using the graphical design view of Access show the full name, birthdate and province of employees who were born in the 1970s, 1980s or 1990s in Alberta.
- TA will provide 5 minutes for you to form the graphical query

Author: James Tam

Student Exercise #7B: Query With Multiple Criteria

- Form the previous query using SQL
- TA will provide 5 minutes

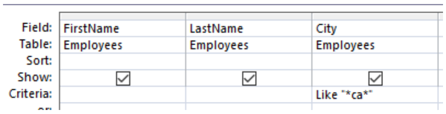
Author: James Tam

Week7: Second Tutorial

- Using wildcards in queries
- Forming multi-table queries
- Defining calculated values as table attributes and during the query
- Calculated values: attributes of a table vs. values derived during queries

First Query: Wildcard

- The wildcard * can be used for text searches where only a part of the string is known.
  - Similar to specifying the format string of validation rule quotes must be used to enclose the string and the ‘`like`’ operator must be used.
- Example: Show the first name, last name and city of all employees that live in a city that contains ‘ca’ somewhere in the city name



Field:	FirstName	LastName	City
Table:	Employees	Employees	Employees
Sort:			
Show:	☑	☑	☑
Criteria:			Like "*ca*"

```
SELECT Employees.FirstName, Employees.LastName, Employees.City
FROM Employees
WHERE ((Employees.City Like "*ca*");
```

Author: James Tam

Student Exercise #8

- Using the graphical design view of Access show the first name, last name and birthdate of employees born in the 1980s
- Form the equivalent query using SQL
- TA provides ~5 minutes to complete the exercise

Author: James Tam

Student Exercise #9

- Using the graphical design view of Access show the first name of all employees whose last name begins either of the following letters ‘s’, ‘c’
- Form the equivalent query using SQL
- TA provides 5 minutes to complete the exercise

Author: James Tam

Multi-Table Queries

- Information for the query comes from two or more tables
- Unlike a single table query:

```
SELECT Table.Attribute…
From Table
```

- The point of overlap between the tables must be specified
  - This is the primary-foreign relationship
  - Example:
    - ON Provinces.ProvinceCode = Employees.Province
  - Also the multi-table aspect of the query needs to be specified
  - Example
    - `Provinces INNER JOIN Employees`

Author: James Tam

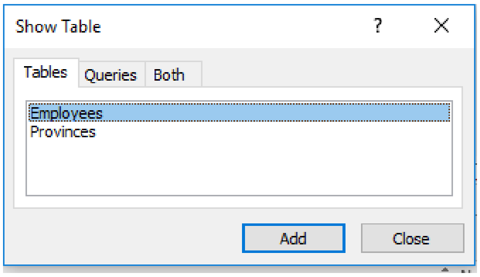
Query 2A: Multi-Table Query (Simple Query, Graphical Design View Version)

- Show the employee number (Employees table) and province code (Provinces table) of the employees

Author: James Tam

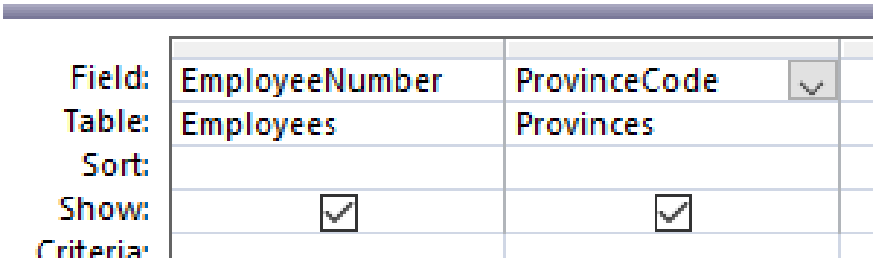
Step 1: Select The Tables

- Same as with single table query but select more than one table



Author: James Tam

Step 2: Select The Appropriate Attributes From The Tables



Field:	EmployeeNumber	ProvinceCode
Table:	Employees	Provinces
Sort:		
Show:	☑	☑
Criteria:		

Query 2: Multi-Table Query (Simple Query, SQL)

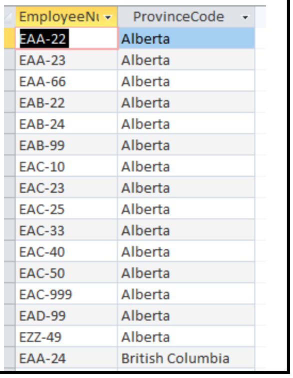
- SELECT:
  - Same approach as single table query (`table.attribute`)
  - `SELECT Employees.EmployeeNumber, Provinces.ProvinceCode`
- FROM:
  - Remember:
    - The multi-table aspect of the query requires an “INNER JOIN” to connect tables
    - `Provinces INNER JOIN Employees`
    - The point of overlap requires an “ON” for the primary-foreign keys to be specified
    - `ON Provinces.ProvinceCode = Employees.Province`

Query 2: Multi-Table Query (Simple Query, SQL)

- Complete query:

```
SELECT Employees.EmployeeNumber, Provinces.ProvinceCode
FROM Provinces INNER JOIN Employees
  ON Provinces.ProvinceCode = Employees.Province;
```

Results



EmployeeN	ProvinceCode
EAA-22	Alberta
EAA-23	Alberta
EAA-66	Alberta
EAB-22	Alberta
EAB-24	Alberta
EAB-99	Alberta
EAC-10	Alberta
EAC-23	Alberta
EAC-25	Alberta
EAC-33	Alberta
EAC-40	Alberta
EAC-50	Alberta
EAC-999	Alberta
EAD-99	Alberta
EZZ-49	Alberta
EAA-24	British Columbia

Author: James Tam

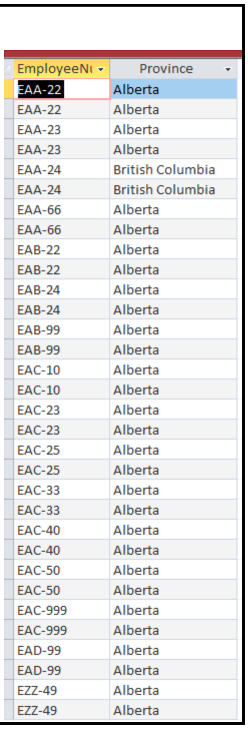
Query 2B: Incorrect SQL Version

- What if the multi-table nature of the query and the point of overlap is not specified.
- That is, the single table format is used:
  - `SELECT Table1.Attribute, Table2.Attribute…`
  - `FROM Table1, Table2…`

Author: James Tam

Query 2B: Incorrect SQL Version

```
SELECT Employees.EmployeeNumber,
Employees.Province
FROM Provinces, Employees;
```



EmployeeNı	Province
EAA-22	Alberta
EAA-22	Alberta
EAA-23	Alberta
EAA-23	Alberta
EAA-24	British Columbia
EAA-24	British Columbia
EAA-66	Alberta
EAA-66	Alberta
EAB-22	Alberta
EAB-22	Alberta
EAB-24	Alberta
EAB-24	Alberta
EAB-99	Alberta
EAB-99	Alberta
EAC-10	Alberta
EAC-10	Alberta
EAC-23	Alberta
EAC-23	Alberta
EAC-25	Alberta
EAC-25	Alberta
EAC-33	Alberta
EAC-33	Alberta
EAC-40	Alberta
EAC-40	Alberta
EAC-50	Alberta
EAC-50	Alberta
EAC-999	Alberta
EAC-999	Alberta
EAD-99	Alberta
EAD-99	Alberta
EZZ-49	Alberta
EZZ-49	Alberta

Author: James Tam

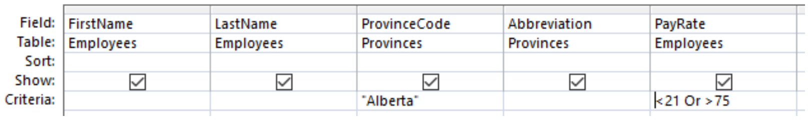
Student Exercise #10

- Define the query (using the graphical design view of Access) as well as using SQL that will display the following information.
  – The full (first and last) name of Employees
  – Two letter abbreviation of the province where the employee lives
  – TA will provide 5 minutes for you to complete the exercise

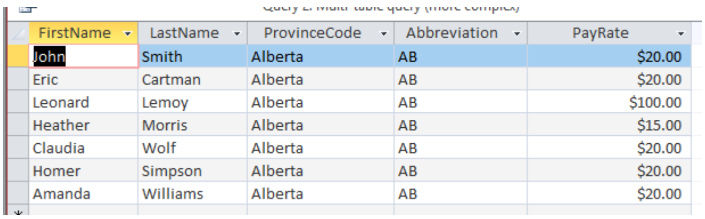
Author: James Tam

Query 3: Multi-tables Query Combined With Logic (Graphical Design View)

- Show full name, province and province abbreviation (both from ‘Provinces’) of employees as well as the pay rate from Alberta employees whose earnings falls within one the following categories: less than $21, more than $75



Field:	FirstName	LastName	ProvinceCode	Abbreviation	PayRate
Table:	Employees	Employees	Provinces	Provinces	Employees
Sort:					
Show:	☑	☑	☑	☑	☑
Criteria:			"Alberta"		<21 Or >75



FirstName	LastName	ProvinceCode	Abbreviation	PayRate
John	Smith	Alberta	AB	$20.00
Eric	Cartman	Alberta	AB	$20.00
Leonard	Lemoy	Alberta	AB	$100.00
Heather	Morris	Alberta	AB	$15.00
Claudia	Wolf	Alberta	AB	$20.00
Homer	Simpson	Alberta	AB	$20.00
Amanda	Williams	Alberta	AB	$20.00

Author: James Tam

Query 3: Multi-tables Query Combined With Logic (SQL)

```
SELECT Employees.FirstName, Employees.LastName,
Provinces.ProvinceCode, Provinces.Abbreviation,
Employees.PayRate

FROM Provinces INNER JOIN Employees ON
Provinces.ProvinceCode = Employees.Province

WHERE ((Provinces.ProvinceCode="Alberta") AND
(Employees.PayRate<21) Or (Employees.PayRate)>75);
```

Author: James Tam

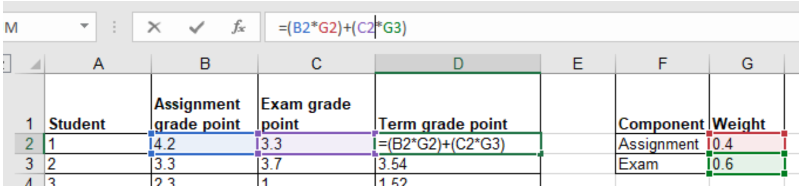
Student Exercise #11

- Define the query (using the graphical design view of Access) as well as using SQL that will display the following information.
  - The full (first and last) name of Employees
  - (The full name) of the province where the employee resides
  - Show only cases where the employee was born on or after March 15, 1967 but before the year 2000.
  - Also only show employees who live either in Calgary or Silent Hill
  - Due to the complexity of this exercise the TA will provide 10 minutes for you to complete the exercise

Author: James Tam

Calculated Values

- Similar to spreadsheet cells that derive their results based on the contents of another cell.
- Example: Cell D2 is derived from the current values from 4 other cells



=(B2*G2)+(C2*G3)

	A	B	C	D	E	F	G
1	Student	Assignment grade point	Exam grade point	Term grade point		Component	Weight
2	1	4.2	3.3	=(B2*G2)+(C2*G3)		Assignment	0.4
3	2	3.3	3.7	3.54		Exam	0.6
4	3	2.3	1	1.52			

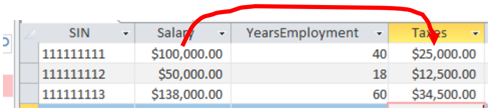
- Example database (New): “`calculated.accdb`”

Author: James Tam

Calculated Values: Table Attributes

- Example (from the database):
  - Taxes paid is equal to one quarter of the employee’s current salary.
  - Similar to the design rules with spreadsheets if values in a database can be derived from another attribute then it should not be manually entered.

Taxes = Salary x 25%

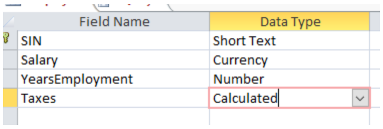


SIN	Salary	YearsEmployment	Taxes
111111111	$100,000.00	40	$25,000.00
111111112	$50,000.00	18	$12,500.00
111111113	$138,000.00	60	$34,500.00

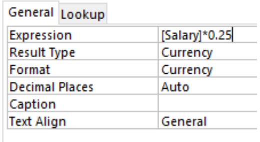
Author: James Tam

Reminder From Lecture: Defining Calculated Attributes

- To specify that a table attribute will be derived, in design view specify the type as ‘calculated’
- (Used when the result applies to all/most records (e.g. all employees must pay taxes) so a column stores taxes owed for each employee.



Field Name	Data Type
SIN	Short Text
Salary	Currency
YearsEmployment	Number
Taxes	Calculated

- The enter the expression (calculation) in the following format:
  - [<Attribute>] or constant <operation> [<Attribute>] or
  - 

Expression	[Salary]*0.25
Result Type	Currency
Format	Currency
Decimal Places	Auto
Caption	
Text Align	General

Author: James Tam

Calculated Values: Derived Only During The Query

- The data is not stored in the columns (attributes) of the table
- Instead the calculation only occurs (the value is generated) on an "as needed" basis during the query
- Example: Cost of paying employees a one time lump sum to retire early
  - This cost should be less than the cost of continuing to pay the employee's salary.
  - Therefore it should only be applied to employees who will actually retire shortly

Author: James Tam

Reminder From Lecture: Deriving Values During The Query (SQL)

Specifying Calculated Values (SQL)

Format:

```
SELECT <Table name.Attribute name1>,
<Table name.Attribute name2>, ...
<[Attribute name] or constant> <expression>
<[Attribute name] or constant> <expression> ...
<[Attribute name] or constant> AS <Query column name>
<Table name.Attribute namen>
```

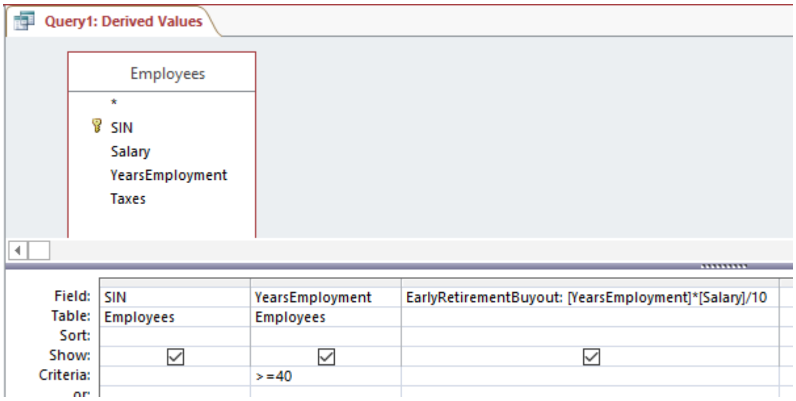
Example:

- SELECT Games.Title, Games.HourlyRate, **[HourlyRate]/60 AS RatePerMinute**

Author: James Tam

Deriving Values: Example Database

- Show the early buyout costs for employees who have worked for 40 or more years.
- Buyout cost = (Salary x Years of employment)/10



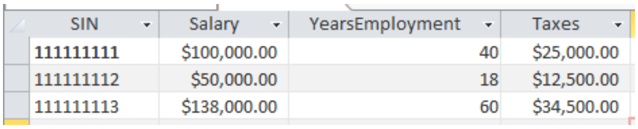
Field:	SIN	YearsEmployment	EarlyRetirementBuyout: [YearsEmployment]*[Salary]/10
Table:	Employees	Employees	
Sort:			
Show:	☑	☑	☑
Criteria:		>=40	
or:			

Author: James Tam

Results Of The Query: Example Database

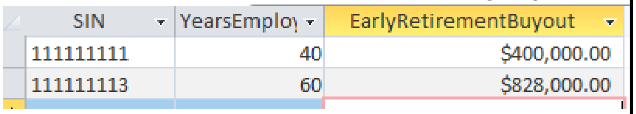
- (Employees who worked 40+ years)
- `EarlyRetirementBuyout: [YearsEmployment]*[Salary]/10`

Employees



SIN	Salary	YearsEmployment	Taxes
111111111	$100,000.00	40	$25,000.00
111111112	$50,000.00	18	$12,500.00
111111113	$138,000.00	60	$34,500.00

Query results

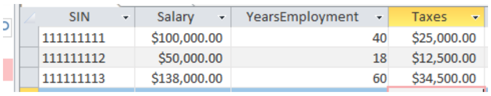


SIN	YearsEmploy	EarlyRetirementBuyout
111111111	40	$400,000.00
111111113	60	$828,000.00

Author: James Tam

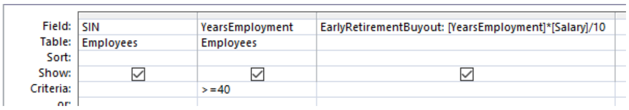
Calculated Values: Which Way?

- Should the value be derived and stored as an attribute in the table or should it be only be derived as the query is run?
- Rule of thumb:
  - Data applies to most/all records: store the result in the attributes of the table (e.g. taxes owed for each employee)



SIN	Salary	YearsEmployment	Taxes
111111111	$100,000.00	40	$25,000.00
111111112	$50,000.00	18	$12,500.00
111111113	$138,000.00	60	$34,500.00

  - Data applies to only some of the records: calculate the results only when the query is run (query conditions may limit which records on which the calculation will be performed)



Field:	SIN	YearsEmployment	EarlyRetirementBuyout: [YearsEmployment]*[Salary]/10
Table:	Employees	Employees	
Sort:			
Show:	☑	☑	☑
Criteria:		>=40	
or:			

Author: James Tam

Calculated Values: Which Way?

- What approach to take on the assignment?
- Read the specifications for this years assignment, the instructions will provide strong clues (or even provide explicit instructions)
- **If you do not follow the instructions specified in the assignment** then you will get few (if any credit) for the appropriate feature.

Author: James Tam