King Fahd University of Petroleum and Minerals

###### Information and Computer Science Department

**ICS 102: Introduction to Computer Programming**

**Summer Semester 2019-2020 (Term 191)  
Homework #1  
[Posted: Wensday October 9th 2019]  
[Due Date: Saturday October 26th 2019 @ 11:59 PM (Midnight)]**

**Submission Guidelines:**

Submit a zipped file containing the following files:

* Q1.java (Java source file) containing your answer to the programming question no. 1.
* Q2.java (Java source file) containing your answer to the programming question no. 2.

PLEASE DO NOT INCLUDE .class FILES IN YOUR SUBMISSION

The zipped file should be named as follows:

**HW1\_XXXXXXXXX\_YourFamilyName.zip**

where:

XXXXXXXXX is your 9 digit KFUPM ID.

YourFamilyName is your family name

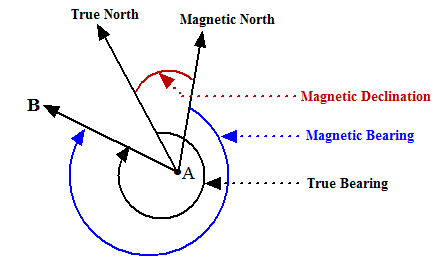
Submission should be made through your ICS 102 Blackboard course page under **Assignments** submission link.

**Important Notes:**

* **Cheating is taken seriously**. Any cheating attempt will result in an F grade in the course.
* **EACH STUDENT IS REQUIRED TO DO THE HOMEWORK ALONE**. COPYING FROM ANY SOURCE IS REGARDED AS CHEATING.
* **Submission link will be available until 9:00am for late submission without penalty.**
* **Submissions via email are not accepted and will be simply ignored**.
* Submission of the homework solution should be in a zipped filed with the format specified above. **Any different formatting/naming will result in reducing the total homework score by half!**
* **You must use proper indentation and meaningful variable names in your programs.**

**Question 1.**

The true bearing from a point A to a point B is the angle measured in degrees, in a clockwise direction, from the line joining the true north and point A to the line joining point A and point B:



The magnetic bearing from a point A to a point B is the angle measured in degrees, in a clockwise direction, from the line joining the magnetic north and point A to the line joining point A and point B.

The magnetic declination of a point A is the angle between the line joining point A to the true north and the line joining point A and the magnetic north.

Note: Magnetic declination is positive for easterly declinations, and negative for westerly declinations.

|  |  |
| --- | --- |
|  |  |

The formulas to find the true bearing of a point B from point A are:

* x = sin(Math.toRadians(longitudeB – longitudeA))
* y = cos(latitudeARadians)\*tan(latitudeBRadians) - sin(latitudeARadians)\*cos((longitudeB – longitudeA) Radians)
* radians = tan-1(x/y);
* angleInDegrees = Math.toDegrees(radians);

Depending on the signs of x and y, the true bearing angle angleInDegrees computed above is normalized as:

If x = 0 and y > 0 and magneticDeclination > 0: angleInDegrees = angleInDegrees + 360

If x = 0 and y < 0: angleInDegrees = angleInDegrees + 180

If x > 0 and y < 0: angleInDegrees = angleInDegrees + 180

If x < 0 and y < 0: angleInDegrees = angleInDegrees + 180

If x < 0 and y > 0: angleInDegrees = angleInDegrees + 360

To find the magnetic bearing (or direction) of a point B from a point A, use the following formula:

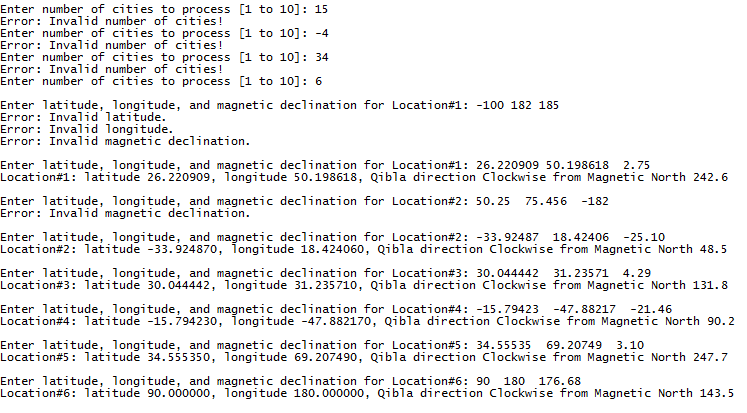
magneticBearing = normalizedTrueBearing - magneticDeclination

Write a Java program that prompts for and reads the number N of cities or locations to be processed. It then loops N times to prompt for and read, for each location, the decimal latitude, decimal longitude, and decimal magnetic declination. It then computes and displays, for each location, the Qibla direction (or bearing) from Magnetic North.

Note:

* Latitude of Al-Kaabah is 21.422506 and its longitude is 39.826203 [use point B as Al-Kaabah in the above formulas]
* N can have values from 1 to 10 inclusive, i.e, -1 ≤ N ≤ 10
* A decimal latitude can have values from -90 degrees to 90 degrees inclusive, i.e, -90 ≤ degrees ≤ 90
* A decimal longitude can have values from -180 degrees to 180 degrees inclusive, i.e., -180 ≤ degrees ≤ 180
* A decimal magnetic declination can have values from -180 degrees to 180 degrees inclusive, i.e., -180 ≤ degrees ≤ 180
* The location of point A must be different from that of point B.
* Your program must recover from input errors in a manner indicated in the sample program run below.
* Your program must use appropriate constants

Sample Program run:



You may use the following data to test your program:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Location | Latitude | Longitude | Magnetic declination on 22 October 2016 | Bearing from True North | Bearing From Magnetic North |
| Al-Khobar, KSA | 26.220909 | 50.198618 | 2.75 | 245.3 | 242.6 |
| Cape Town, South Africa | -33.92487 | 18.42406 | -25.10 | 23.4 | 48.5 |
| Cairo, Egypt | 30.04442 | 31.23571 | 4.29 | 136.1 | 131.8 |
| Brasilia, Brazil | -15.79423 | -47.88217 | -21.46 | 68.8 | 90.2 |
| Kabul, Afghanistan | 34.55535 | 69.20749 | 3.10 | 250.8 | 247.7 |
| **North Pole** | **90** | **180** | **176.68** | **320.2** | **143.5** |
| Tabuk, KSA | 28.402385 | 36.572671 | 4.21 | 156.4 | 152.2 |
| Toronto, Canada | 43.65323 | -79.38318 | -10.48 | 54.6 | 65.1 |
| New York, USA | 40.71278 | -74.00594 | -12.93 | 58.48 | 71.4 |
| Tokyo, Japan | 35.70903 | 139.73199 | -7.42 | 293.0 | 300.5 |
| Melbourne, Australia | -37.81363 | 144.96303 | -11.60 | 278.8 | 290.4 |

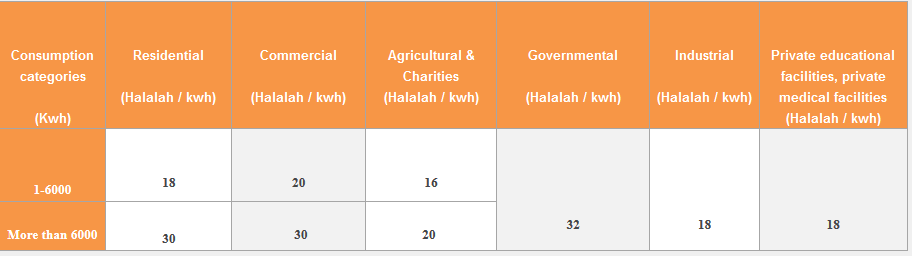
Note: The computations may not be correct at the poles.

**Question 2.**

Write a program in Java to compute the total power (in Kilo Watts) and cost (in SR) related to the power distributed to the power recipients from a power station. Your program will compute the power and cost of each recipient using the following formulas:

**Power (in Watts) = I (current in Amps) × V (voltage in Volts)**

You will need to use the following table that is taken from the Electricity Company (actual numbers may be different), in order to compute the cost for computing electricity consumed by variant recipients:



The current and voltage values will be inserted by the user and they should be greater than zero or equal to -1 (which is explained below). Otherwise, you will display an invalid input error message. Note that the power consumption in the Electricity Company table (above) is in Kilo Watts Hour (Kwh), and that you need not be concerned with the Hour part because the Kilo Watts are computed over the full period of one full month, and therefore, the hours are factored in the computation. In other words, you should just be concerned with computing how many Kilo Watts were consumed by a certain recipient. For example, if a house consumed 6000 Kilo Watts then they will pay 6000 \* 0.18 = 1080 SR. If they consumed 7000 Kilo Watts then they will pay 1080 + (1000 \* 0.3) = 1380 SR.

Here are sample data that you need to use to run your program. Remember that your program should be general for any set size of data.

|  |  |  |
| --- | --- | --- |
| Customer Recipient | Voltage (volts) | Current (Amps) |
| Mall | 220 | 1770000 |
| House | 220 | 2500 |
| Government University | 110 | 2880000 |
| Mosque | 220 | 500 |
| Hyper Market | 240 | 570000 |
| Airport | 240 | 12800000 |
| Private Hospital | 220 | 99000 |

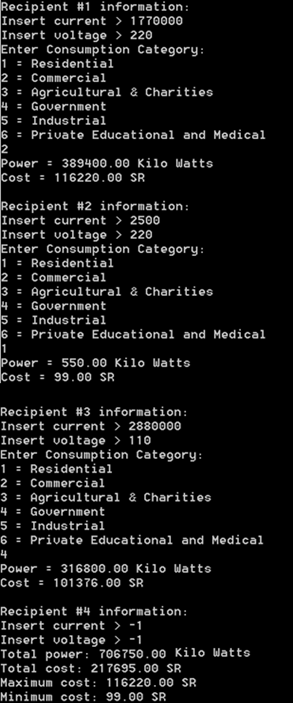
Your program should be general for any number of recipients and should continue computing the power and cost until the user inserts -1 for voltage and/or current. Once the program completes computing the power and cost for all recipients, it should display the following information:

* Total power
* Total cost
* Highest and lowest cost

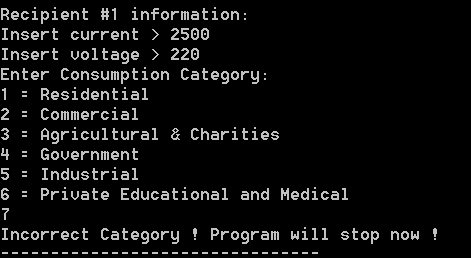
Note: **Programs that do not use repetition will be penalized heavily.**

The sample output, below, shows the computation of the top 3 samples in the table. HOWEVER, our graders will test your code for all other entries in the table so you need to make sure your program runs for all possible cases listed in the table.

**Sample output 1:**



**Sample output 2:**



**Sample output 3:**

