CS 200 Lab 08 Spring 2018

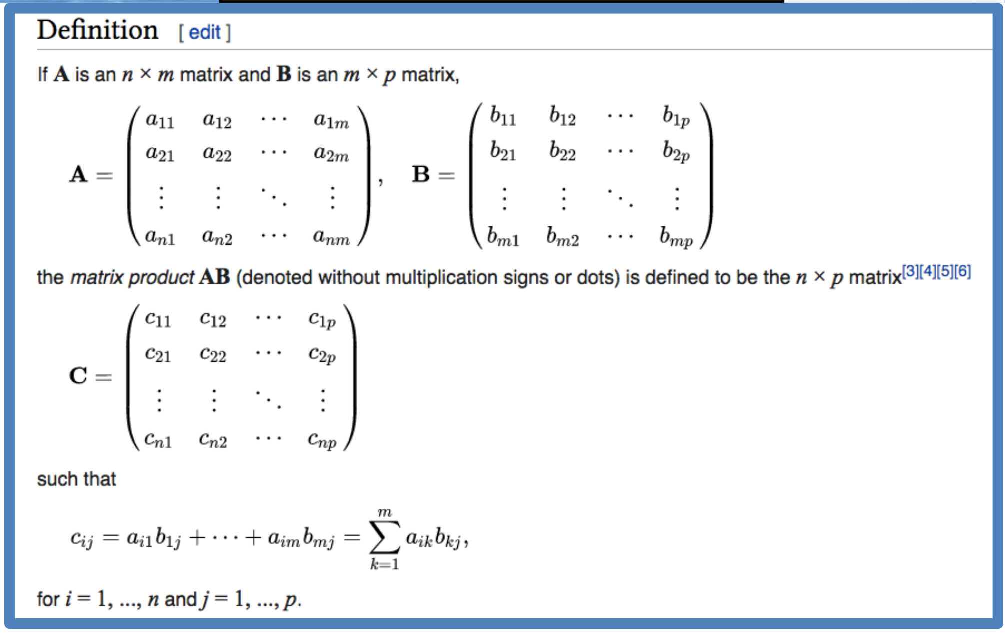
Introduction to Programming

(CS200)

Shafay Shamail

Programming Practice

BJARNE STROUSTRUP
Programming
Principles and Practice Using C++
I will not write any more bad code
I will not write any more bad code
I will not write any more bad code
I will not write any more bad code
I will not write any more bad code
I will not write any more bad code
I will not write any more bad code
I will not write any more bad code
I will not write any more bad code
I will not write any more bad code
I will not write any more bad code
The Practice of Programming
Brian W. Kernighan
Rob Pike
Simplicity
Clarity
Generality
ADDISON-WESLEY PROFESSIONAL COMPUTING SERIES



Definition [edit]

If $\mathbf{A}$ is an $n \times m$ matrix and $\mathbf{B}$ is an $m \times p$ matrix,

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1p} \\ b_{21} & b_{22} & \cdots & b_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ b_{m1} & b_{m2} & \cdots & b_{mp} \end{pmatrix}$$

the *matrix product* $\mathbf{AB}$ (denoted without multiplication signs or dots) is defined to be the $n \times p$ matrix[3][4][5][6]

$$\mathbf{C} = \begin{pmatrix} c_{11} & c_{12} & \cdots & c_{1p} \\ c_{21} & c_{22} & \cdots & c_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ c_{n1} & c_{n2} & \cdots & c_{np} \end{pmatrix}$$

such that

$$c_{ij} = a_{i1}b_{1j} + \cdots + a_{im}b_{mj} = \sum_{k=1}^{m} a_{ik}b_{kj},$$

for $i = 1, ..., n$ and $j = 1, ..., p$.

https://en.wikipedia.org/wiki/Matrix_multiplication

CS 200 Lab 08 Spring 2018

Lab Guidelines

1. Make sure you get your work graded before the lab time ends.
2. You put all your work onto the LMS folder designated for the lab (i.e. “Lab05”) before the time of the lab ends.
3. Talking to each other is NOT permitted. If you have a question, ask the lab assistants.
4. The object is not simply to get the job done, but to get it done in the way that is asked for in the lab.
5. Any cheating case will be reported to Disciplinary Committee without any delay.

NOTE: Define a class interface separately and its methods separately. Do not write inline code.

Marks: Name: ____________________ Roll #: ____________________

Task 1	1	2	3	4	5			Total
	10	10	10	10	10			50

Task 2	1	2	3	4	5			Total
	10	10	10	10	10			50

Let’s Begin……

Total marks Obtained

/100

CS 200 Lab 08 Spring 2018

Task 1: (50)

In this lab you will be writing operations for matrix manipulations: add, subtract, multiply.
The program will prompt the user to enter the name of the files to be read for matrix A and matrix B respectively. It will then perform addition, subtraction, and multiplication of these two matrices and then print the Matrix A, Matrix B and the results of addition, subtraction, and multiplication. Before performing an operation the function must verify whether it is possible to execute that operation or not and generate appropriate error message if the operation is not possible.

The functions are:

1. readMatrix with void return and one 2D array argument and one string argument. 10
2. printMatrix with void return and one 2D array argument. 10
3. sumMatrix with void return and three 2D array arguments. 10
4. diffMatrix with void return and three 2D array arguments. 10
5. multiMatrix with void return and three 2D array arguments. 10

- **STOP AND SHOW YOUR WORK TO THE TA**

Task 2: **(50)**

Develop a program that builds a Phonebook system. You have to do all of this with the help of a class named Contact, and a class named Phonebook.

The attributes:

A. Contact

   a. Name (type: string)
   b. Number(type: double)
   c. Relation(type: string)
   d. Phonebook_number(type:int) //this tells which phone book the contact belongs to

   Note: These data members should be private, not public.

B. Phonebook

   a. Name (type: string)
   b. Contacts (type: array containing objects of type: Contact)
   c. totalContactCount (type: int)

   Note: These data members should be private too.

Keep in mind that you need to write get and set member-functions for each primitive attribute (i.e. Name, Number, Relation etc.).

Your system needs to

1. Add a new contact to the existing phonebook (let the user then input the attributes of the contact) 10
2. View a contact i.e. the user identifies the contact by its number 10
3. Delete a contact i.e. the user identifies the contact by its number 10
4. View all contacts (with all their attributes) 10
5. View the number of contacts in the phonebook 10

STOP AND SHOW YOUR WORK TO THE TA

Zip your tasks into one folder with format:

YourRollNo-Lab07

example "**2001001-Lab07**" and upload on LMS before the tab is closed. You will not be given extra time.