

CS 2413 Data Structures – Spring 2016 – Programming Project 3

Due March 7, 2016 – 11:59 PM

Objectives

1. [50 Points] Create all the required data structures along with the implementation of each of the following methods for all classes. **You are required to demonstrate the working of the following methods for each of the methods by invoking them from a main program.**
 - a. empty constructor
 - b. non-empty constructor (sets the initial size of the vector)
 - c. destructor
 - d. copy constructor
 - e. overloaded equal to operator
 - f. ostream operator
2. [20 Points] Read the redirected input and create all the data structures.
3. [20 Points] Demonstrate the working of the classes with two different data types: int and character strings.
4. [10 Points] Document your project thoroughly as the examples in the textbook. This includes but not limited to header comments for all classes/methods, explanatory comments for each section of code, meaningful variable and method names, and consistent indentation.

Project Description

A Cell is a linked list node and can contain any data (in this project we will assume that they are all integers). A Cell Node contains a value (a data type) and a pointer to a Cell. The Master Cell contains an array of Cell Nodes.

The input file contains an unknown number of lines of input. Each line contains an integer (we will call it info) and followed by an integer (call it noltems). Following this integer, there will be several integers (the number of them equal to noltems). For example, let us say that the input line is as follows:

106 5 1200 1800 300 4999 5000

106 is the info, noltems is 5 and the 5 integers are 1200, 1800, 300, 4999, and 5000. When you read the line, you will create a CellNode object store the value 106 in its _info field. You will create a linked list with the 5 values and store the pointer to the first element of the linked list created in the field _myCell of the CellNode object. The CellNode object that was created is stored in the first element of the vector _myCellNodes of the MasterCell object created in the main program.

106, 5 1200 1800 300 4999 5000
90, 3 30 1000 80
200, 3 20 10 40 90
50, 2 41 31
60, 3 88 75 26
65, 1 51

The diagram illustrates a linked list structure. A large oval on the left is labeled "Master Cell". Inside this oval is a vertical column of six cells, each containing a number and an arrow pointing to the right. The numbers are 106, 90, 200, 50, 60, and 65. The cell containing 65 is also circled. An arrow points from this circled cell to a box labeled "Cell Node". Another arrow points from the "Master Cell" oval to a box labeled "Master Cell". To the right of the "Master Cell" column, there are several other cells, each containing a number and an arrow pointing to the right. These cells are arranged in a way that suggests a linked list structure. The numbers in these cells are 1200, 30, 20, 41, 88, 51, 1800, 1000, 10, 31, 75, 26, 300, 80, 40, 90, 4999, and 5000. The cell containing 75 is also circled. An arrow points from this circled cell to a box labeled "Cell Object".

Below is a second set of input lines:

Operating Systems, 5 1200 1800 300 4999 5000

Compiler construction, 3 30 1000 80

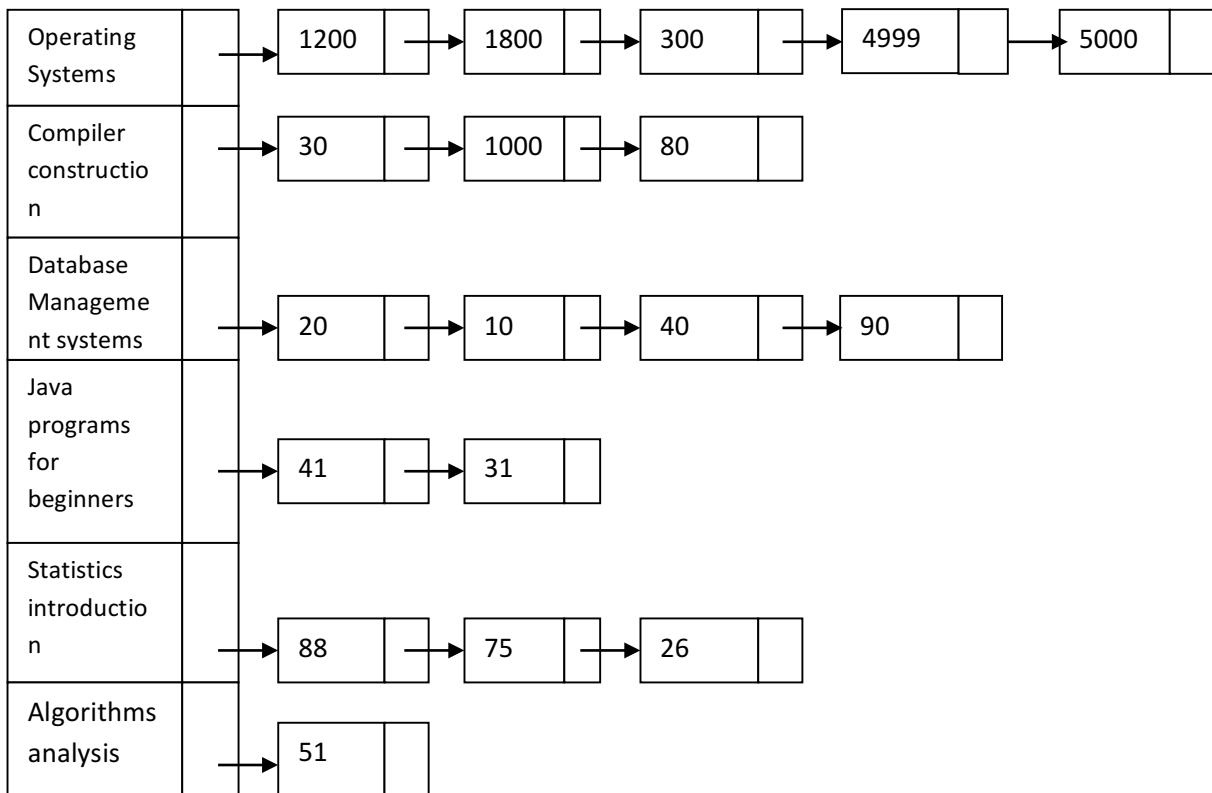
Database Management systems, 3 20 10 40 90

Java programs for beginners, 2 41 31

Statistics introduction, 3 88 75 26

Algorithms analysis, 1 51

The data structure is pictorially depicted below:



After reading all the input and creating appropriate objects, you have to demonstrate the working of all the methods in each of the classes.

Please follow the discussion of this project in class in addition to the description here.

Class Structures

You are required to implement the following class structures along with the implementation of the methods associated with each of them.

```
template <class DT>
class Cell {
protected:
    DT* _value;
    Cell<DT>* _right;
public:
    //All required methods
};

template <class DT1, class DT2>
class CellNode {
protected:
    DT1* _info;
    Cell<DT2>* _myCell;
public:
    //All required methods
};

template <class DT1, class DT2>
class MasterCell {
protected:
    CellNode<DT1,DT2>* _myCellNodes;
public:
    //All required methods
};
```

Constraints

1. In this project, the only header you will use is `#include <iostream>`.
2. None of the projects is a group project. Consulting with other members of this class on programming projects is strictly not allowed and plagiarism charges will be imposed on students who do not follow this.