Computer Programming

Dr. Deepak B Phatak
Dr. Supratik Chakraborty
Department of Computer Science and Engineering
IIT Bombay

Session: Programming using structures – Part 1

Quick Recap of Relevant Topics

- Brief introduction to object-oriented programming

- Defining structures in C++

- Accessing members of structures

- Initializing and copying structures

Dr. Deepak B. Phatak & Dr. Supratik Chakraborty, IIT Bom

Overview of This Lecture

- Getting our hands dirty
    - C++ programming with structures

Dr. Deepak B. Phatak & Dr. Supratik Chakraborty, IIT Bom

Acknowledgment

- Some examples in this lecture are from

  An Introduction to Programming Through C++

  by Abhiram G. Ranade

  McGraw Hill Education 2014

- All such examples indicated in slides with the citation
  AGRBook

Dr. Deepak B. Phatak & Dr. Supratik Chakraborty, IIT Bombay

Recall: Library Information Management System [Ref. AGRBook]

- Every patron has a numerical id
- Every book has an accession number
- **Check out:** A patron can check out upto 3 books at any tim
- **Claim:** If X has not already checked out 3 books, she can claim a book checked out by Y

  When Y returns the book, it is held for X and cannot be le

  to others
- **Return:** A patron can return a book checked out by her at any time

Dr. Deepak B. Phatak & Dr. Supratik Chakraborty, IIT Bombay

Recall: Relevant Structures and Arrays

```
struct Book {
    char title[50];
    char authors[500];
    double price;
    int accNum;
    bool checkOutStatus;
    int claimantId;
};
```

Book libraryShelf[1000]

Recall: Relevant Structures and Arrays

```
struct Book {

    char title[50];

    char authors[500];

    double price;

    int accNum;

    bool checkOutStatus;

    int claimantId;

};
```

Book libraryShelf[1000]

Assume checkOutStatus and claimantId of all elements of arr libraryShelf initialized to "false" and "-1" respectively

Dr. Deepak B. Phatak & Dr. Supratik Chakraborty, IIT Bombay

Recall: Relevant Structures and Arrays

```
struct Patron {
    char name[50];
    char address[100];
    int uniqueId;
    int numBooksChkdOu
    int claimdBookAccNur
};
```

Patron libraryPatrons[20(

Recall: Relevant Structures and Arrays

Assume numBooksChkdOut and claimdBookAccNum for all elements of array libraryPatrons initialized to "0" and "-1" respectively

```
struct Patron {
    char name[50];
    char address[100];
    int uniqueId;
    int numBooksChkdOu
    int claimdBookNum
};
```

Patron libraryPatrons[200

C++ Function for Checking Out a Book

currPatron (of type **Patron**) to check out currBook (of type **Bo**

- Check if currPatron has already checked out 3 books
  - If so, print appropriate message and return
  - Otherwise,
    * If currBook is already checked out, print appropriate messag
      and return
    * Otherwise, if currBook not already claimed by a different pat
      - Increment value of currPatron.numBooksChkdOut
      - Set currBook.checkOutStatus of book to true

C

We want changes to members of currPatron an...
currBook to persist after currBook is checked ou...
Need functions with parameters passed by
reference

* Otherw..., if currBook n... already claimed by a different patr...

- Increm...nt value of currPatron.numBooksChkdOut

- Set currBook.checkOutStatus of book to true

C++ Function for Checking Out a Book

```cpp
void checkOutBook(Patron &currPatron, Book &currBook)
{
        … Code for checking out a book …
}


int main() {
        … Other code …
        checkoutBook(libraryPatrons[i], libraryShelf[j]);
        … Other code …
}
```

C++ Function for Checking Out a Book

// PRECONDITION: Members of currPatron and currBook are
//   properly initialized – no garbage values
void checkOutBook(Patron &currPatron, Book &currBook)
{

    … Code for checking out a book …

}
// POSTCONDITION: If currBook is lent to currPatron, members
// currPatron and currBook appropriately updated

Dr. Deepak B. Phatak & Dr. Supratik Chakraborty, IIT Bombay

C++ Function for Checking Out a Book

```cpp
void checkOutBook(Patron &currPatron, Book &currBook)
{
    if (currPatron.numBooksChkdOut < 3) {
        ... Code for checking out a book (part 1) ...
    }
    else { cout << "Sorry! Three books have already been checked";
        cout << " out by " << currPatron.name << endl;
        return;
    }
}
```

C++ Function for Checking Out a Book

```cpp
if (currPatron.numBooksChkdOut < 3) {
    if (currBook.checkOutStatus == true) {
        cout << "Sorry!  Book " << currBook.title;
        cout << " (Accession # " << currBook.accNum << ") ";
        cout << " already checked out!" << endl;
        return;
    }
    else {  … Code for checking out a book (part 2) … }
}
```

C++ Function for Checking Out a Book

```
else {
   if ((currBook.claimantId != -1) &&
      (currBook.claimantId != currPatron.uniqueId)) {
      cout << "Sorry! There is already a pending claim on book ";
      cout << currBook.title << " (Acc # " << currBook.accNum << ") ";
      cout << " by a different patron." << endl;
      return;
   }
   else { … Code for checking out a book (part 3) … }
}
```

C++ Function for Checking Out a Book

```
else {
  currBook.checkOutStatus = true;
  currPatron.numBooksChkdOut ++;
  if (currBook.claimantId ==
      currPatron.uniqueId) {
    currPatron.claimdBookAccNum = -1;
    currBook.claimantId = -1;
  }
  return;
}
```

currBook, currPatron **passed by reference**

Therefore, structures used as parameters i the calling function a updated

Summary

- Programming using structures
    - Passing structures as parameters to functions
    - Accessing members of structures in program

We aren't done yet with our implementation of differe
functions needed in the library information managem
system. Subsequent lecture to cover these.

Dr. Deepak B. Phatak & Dr. Supratik Chakraborty, IIT Bombay