**USER**

**FUNCTION USER(id: int, name: string, email: string, pass: string, type: string)**

SET userID = id

SET name = name

SET email\_address = email

SET pass\_word = pass

SET account\_type = type

END FUNCTION

login(email, pass): bool

**FUNCTION login(email: string, pass: string) RETURNS bool**

IF email == email\_address AND pass == pass\_word THEN

RETURN true

ELSE

RETURN false

END IF

END FUNCTION

getAccType(): string

**FUNCTION getAccType() RETURNS string**

RETURN account\_type

END FUNCTION

getUserID(): int

**FUNCTION getUserID() RETURNS int**

RETURN userID

END FUNCTION

getEmail(): string

**FUNCTION getEmail() RETURNS string**

RETURN email\_address

END FUNCTION

viewInfo()

**FUNCTION viewInfo()**

PRINT "User ID: " + userID

PRINT "Name: " + name

PRINT "Email: " + email\_address

PRINT "Account Type: " + account\_type

END FUNCTION

**Class: ADMIN**

**FUNCTION ADMIN(id: int, name: string, email\_address: string, pass\_word: string)**

CALL USER(id, name, email\_address, pass\_word, "Admin")

END FUNCTION

**FUNCTION view\_Users(user\_list: vector<USER\*>&)**

FOR EACH user IN user\_list

CALL user.viewInfo()

PRINT "it i hope it prints"

END FOR

END FUNCTION

**FUNCTION viewBooks(book\_list: vector<Book\*>&)**

FOR EACH book IN book\_list

CALL book.SHOW()

PRINT "the books :3"

END FOR

END FUNCTION

**Class : Librarian**

**FUNCTION LIBRARIAN(id: int, name: string, email\_address: string, pass\_word: string)**

CALL USER(id, name, email\_address, pass\_word, "Librarian")

SET books\_list = empty vector<Book\*>

SET RESERVED\_BOOKS = empty vector<int>

END FUNCTION

**FUNCTION addBook(bookId: int, title: string)**

FOR EACH book IN books\_list

IF book.getBookID() == bookId THEN

PRINT "Error: Book ID " + bookId + " already exists"

RETURN

END IF

END FOR

CREATE new\_book = new Book(bookId, title, "Available")

ADD new\_book TO books\_list

PRINT "Book added successfully: ID " + bookId + ", Title: " + title

END FUNCTION

**FUNCTION remove\_book(bookId: int)**

FOR i FROM 0 TO books\_list.size() - 1

IF books\_list[i].getBookID() == bookId THEN

DELETE books\_list[i]

REMOVE books\_list[i] FROM books\_list

PRINT "Book removed successfully: ID " + bookId

RETURN

END IF

END FOR

PRINT "Error: Book ID " + bookId + " not found"

END FUNCTION

**FUNCTION editBookStatus(bookId: int, newStatus: string)**

FOR i FROM 0 TO books\_list.size() - 1

IF books\_list[i].getBookID() == bookId THEN

CALL books\_list[i].editStatus(newStatus)

PRINT "Book status updated: ID " + bookId + " to " + newStatus

RETURN

END IF

END FOR

PRINT "Error: Book ID " + bookId + " not found"

END FUNCTION

**FUNCTION ShowBooks()**

IF books\_list IS EMPTY THEN

PRINT "No books available"

RETURN

END IF

FOR EACH book IN books\_list

CALL book.SHOW()

PRINT "The book list goes here"

END FOR

END FUNCTION

**FUNCTION generateReport()**

SET totalBooks = books\_list.size()

SET availableBooks = 0

SET borrowedBooks = 0

FOR EACH book IN books\_list

IF book.getStatus() == "Available" THEN

INCREMENT availableBooks

ELSE IF book.getStatus() == "Borrowed" THEN

INCREMENT borrowedBooks

END IF

END FOR

PRINT "Total Books: " + totalBooks

PRINT "Available Books: " + availableBooks

PRINT "Borrowed Books: " + borrowedBooks

PRINT "Reserved Books: " + RESERVED\_BOOKS.size()

END FUNCTION

**Class : Member**

**FUNCTION MEMBER(id: int, name: string, email\_address: string, pass\_word: string)**

CALL USER(id, name, email\_address, pass\_word, "Member")

SET Borrowed\_Books = empty vector<int>

SET reservedBooks = empty vector<int>

END FUNCTION

**FUNCTION borrow\_book(bookId: int, Librarian: LIBRARIAN)**

IF Borrowed\_Books.size() >= 5 THEN

PRINT "Cannot borrow more than 5 books"

RETURN

END IF

FOR i FROM 0 TO Librarian.books\_list.size() - 1

IF Librarian.books\_list[i].getBookID() == bookId AND Librarian.books\_list[i].getStatus() == "Available" THEN

CALL Librarian.books\_list[i].editStatus("Borrowed")

ADD bookId TO Borrowed\_Books

PRINT "Book borrowed successfully: ID " + bookId

RETURN

END IF

END FOR

PRINT "Error: Book ID " + bookId + " not available or not found"

END FUNCTION

**FUNCTION reserveBook(bookId: int, Librarian: LIBRARIAN)**

IF reservedBooks.size() >= 5 THEN

PRINT "Cannot reserve more than 5 books"

RETURN

END IF

FOR i FROM 0 TO Librarian.books\_list.size() - 1

IF Librarian.books\_list[i].getBookID() == bookId AND Librarian.books\_list[i].getStatus() == "Borrowed" THEN

ADD bookId TO reservedBooks

ADD bookId TO Librarian.RESERVED\_BOOKS

PRINT "Book reserved successfully: ID " + bookId

RETURN

END IF

END FOR

PRINT "Error: Book ID " + bookId + " not borrowed or not found"

END FUNCTION

**FUNCTION return\_book(bookId: int, Librarian: LIBRARIAN)**

FOR i FROM 0 TO Borrowed\_Books.size() - 1

IF Borrowed\_Books[i] == bookId THEN

FOR j FROM 0 TO Librarian.books\_list.size() - 1

IF Librarian.books\_list[j].getBookID() == bookId THEN

CALL Librarian.books\_list[j].editStatus("Available")

REMOVE Borrowed\_Books[i] FROM Borrowed\_Books

PRINT "Book returned successfully: ID " + bookId

RETURN

END IF

END FOR

END IF

END FOR

PRINT "Error: Book ID " + bookId + " not borrowed by this member"

END FUNCTION

**FUNCTION getBorrowedBooksCount() RETURNS int**

RETURN Borrowed\_Books.size()

END FUNCTION

**Class : Book**

**FUNCTION Book(bookID: int, title: string, status: string)**

SET bookID = bookID

SET title = title

SET status = status

END FUNCTION

**FUNCTION getBookID() RETURNS int**

RETURN bookID

END FUNCTION

**FUNCTION getTitle() RETURNS string**

RETURN title

END FUNCTION

**FUNCTION getStatus() RETURNS string**

RETURN status

END FUNCTION

**FUNCTION editStatus(new\_status: string)**

SET status = new\_status

END FUNCTION

**FUNCTION SHOW()**

PRINT "Book ID: " + bookID

PRINT "Title: " + title

PRINT "Status: " + status

END FUNCTION

**Class : Main**

FUNCTION main() RETURNS int

SET Users=empty vector<USER\*>

ADD new LIBRARIAN(1,"example1","email","pass123") TO Users

ADD new ADMIN(2,"example2","email","admin123") TO Users

ADD new MEMBER(3,"Example3","Email","user123") TO Users

SET librarian=CAST Users[0] TO LIBRARIAN

IF librarian.getAccType()!="Librarian" THEN

PRINT "Error: Failed to cast to LIBRARIAN"

FOR EACH user IN Users

DELETE user

END FOR

RETURN 1

END IF

PRINT "Success: Librarian cast successful for "+librarian.getName()

WHILE true

PRINT "1) Create a new user"

PRINT "2) Login"

PRINT "3) Exit"

PRINT "Enter "

INPUT choice

IF choice==1 THEN

PRINT "Create new user"

INPUT name,email,pass,type

SET new\_id=0

FOR EACH user IN Users

INCREMENT new\_id

END FOR

INCREMENT new\_id

IF type=="Admin" THEN

ADD new ADMIN(new\_id,name,email,pass) TO Users

ELSE IF type=="Member" THEN

ADD new MEMBER(new\_id,name,email,pass) TO Users

ELSE IF type=="Librarian" THEN

ADD new LIBRARIAN(new\_id,name,email,pass) TO Users

ELSE

PRINT "please Retype"

CONTINUE

END IF

PRINT "Success: Account created for "+name+" (ID: "+new\_id+")"

ELSE IF choice==2 THEN

PRINT "Login

INPUT email,pass

SET login\_success=false

FOR EACH user IN Users

IF user.login(email,pass) THEN

SET login\_success=true

PRINT "Success: Logged in as "+user.getAccType()+" ("+user.getName()+")"

CALL user.viewInfo()

IF user.getAccType()=="Librarian" THEN

SET librarian=CAST user TO LIBRARIAN

WHILE true

PRINT "Lib menu"

PRINT "1) Add a book"

PRINT "2) Remove a book"

PRINT "3) Edit book status"

PRINT "4) Show all books"

PRINT "5) Generate report"

PRINT "6) Logout"

PRINT "Enter option: "

INPUT opt

IF opt==1 THEN

INPUT book\_id,title

CALL librarian.addBook(book\_id,title)

ELSE IF opt==2 THEN

INPUT book\_id

CALL librarian.remove\_book(book\_id)

ELSE IF opt==3 THEN

INPUT book\_id,new\_status

CALL librarian.editBookStatus(book\_id,new\_status)

ELSE IF opt==4 THEN

CALL librarian.ShowBooks()

ELSE IF opt==5 THEN

CALL librarian.generateReport()

ELSE IF opt==6 THEN

PRINT "Logging out..."

BREAK

ELSE

PRINT "Error: Invalid option"

END IF

END WHILE

ELSE IF user.getAccType()=="Member" THEN

SET member=CAST user TO MEMBER

WHILE true

PRINT "Mem menu"

PRINT "1) View all books"

PRINT "2) Borrow a book"

PRINT "3) Reserve a book"

PRINT "4) Return a book"

PRINT "5) Logout"

PRINT "Enter option: "

INPUT opt

IF opt==1 THEN

CALL librarian.ShowBooks()

ELSE IF opt==2 THEN

INPUT book\_id

CALL member.borrow\_book(book\_id,librarian)

ELSE IF opt==3 THEN

INPUT book\_id

CALL member.reserveBook(book\_id,librarian)

ELSE IF opt==4 THEN

INPUT book\_id

CALL member.return\_book(book\_id,librarian)

ELSE IF opt==5 THEN

PRINT "Logging out..."

BREAK

ELSE

PRINT "Error: Invalid option"

END IF

END WHILE

ELSE IF user.getAccType()=="Admin" THEN

SET admin=CAST user TO ADMIN

WHILE true

PRINT "Admin menu"

PRINT "1) View all users"

PRINT "2) View all books"

PRINT "3) Logout"

PRINT "Enter option: "

INPUT opt

IF opt==1 THEN

CALL admin.view\_Users(Users)

ELSE IF opt==2 THEN

CALL admin.viewBooks(librarian.books\_list)

ELSE IF opt==3 THEN

PRINT "logging out."

BREAK

ELSE

PRINT "wrong number"

END IF

END WHILE

END IF

END IF

END FOR

IF NOT login\_success THEN

PRINT "Login failed"

END IF

ELSE IF choice==3 THEN

PRINT "Exiting "

BREAK

ELSE

PRINT "Error wrong number

END IF

END WHILE

FOR EACH user IN Users

DELETE user

END FOR

PRINT "Program ended."

RETURN 0

END FUNCTION