

# Object Oriented Programming (IGS2130)

## Lab 5

---

**Instructor:**  
**Choonwoo Ryu, Ph.D.**



INHA UNIVERSITY

# OOP Project: Step 02



- Upgrade our non-OOP based bank application (ver 0.1) to version 0.2
  - Create **Account class** instead of using **Account structure**
  - Apply information hiding
  - Use constructor and destructor
  
- Additional changes
  - Use `char *` and dynamic memory allocation for the member variable of the customer name in **Account class**
  - Use pointer array of **account class** to store multiple accounts in the program

To start easier, use a single file in this project.

# OOP Project: Step 02



## Account class definition

```
class Account {
private:
    int m_accID;
    int m_balance;
    char * m_cusName;

public:
    Account(int ID, int balance, char *cname) {
        // implementation required ....
    }
    ~Account() {
        // implementation required ....
    }
    int GetAccID(void) {
        // implementation required ....
    }
    void Deposit(int money) {
        // implementation required ....
    }
    int Withdraw(int money) {
        // implementation required ....
    }
    void ShowAccInfo(void) {
        // implementation required ....
    }
};
```

# OOP Project: Step 02



## ■ Main function and global variables

```
Account *accArr[MAX_ACC_NUM]; // Account array
int accNum = 0;                // # of accounts

int main(void) {
    int choice, i;

    while (1) {
        ShowMenu();
        cout << "Select menu: ";
        cin >> choice;
        cout << endl;

        switch (bank(choice)) {
            case bank::MAKE:
                MakeAccount();
                break;
            case bank::DEPOSIT:
                DepositMoney();
                break;
            case bank::WITHDRAW:
                WithdrawMoney();
                break;
        }
    }
}
```

```
case bank::INQUIRE:
    ShowAllAccInfo();
    break;
case bank::EXIT:
    for (i = 0; i < accNum; i++)
        delete accArr[i];
    return 0;
default:
    cout << "Illegal selection.." << endl;
}
return 0;
}
```

# OOP Project: Step 02



- Account class handling functions
  - Same function prototype but...
  - Must be modified according to the changes in the **Account class** definition.

```
void ShowMenu(void);  
void MakeAccount(void);  
void DepositMoney(void);  
void WithdrawMoney(void);  
void ShowAllAccInfo(void);  
int GetAccIdx(int);
```