

# **Advanced Programming Concepts with C++**

## **CSI 2372**



### **Tutorial # 4**

#### **Selected exercises and midterm exam discussion**

## Exercise 2.41:



Use your Sales\_data class to rewrite the exercises in § 1.5.1 (p. 22), § 1.5.2 (p. 24), and § 1.6 (p. 25). For now, you should define your Sales\_data class in the same file as your main function.

### Solution:

```
#include <string>
#include <iostream>
struct Sales_data {
    std::string bookNo;
    unsigned units_sold = 0;
    double revenue = 0.0;
};
void ex1_20() {
    Sales_data item;
    double price = 0;
    while(std::cin >> item.bookNo >> item.units_sold >> price) {
        item.revenue = item.units_sold * price;
        std::cout << item.bookNo << " " << item.units_sold << " " << item.revenue << std::endl;
    }
}
```



```
int ex1_21(){
    Sales_data d1, d2;
    double price = 0;
    if (std::cin >> d1.bookNo >> d1.units_sold >> price) {
        d1.revenue = d1.units_sold * price;
        if (std::cin >> d2.bookNo >> d2.units_sold >> price) {
            d2.revenue = d2.units_sold * price;
            if (d1.bookNo == d2.bookNo) {
                unsigned totalSold = d1.units_sold + d2.units_sold;
                double totalRevenue = d1.revenue + d2.revenue;
                std::cout << d1.bookNo << " " << totalSold << " " << totalRevenue << " ";
                if (totalSold != 0)
                    std::cout << totalRevenue / totalSold << std::endl;
                else
                    std::cout << "(no sales)" << std::endl;
            } else {
                std::cerr << "Data must refer to the same ISBN!" << std::endl;
                return -1;
            }
        } else {
            std::cerr << "No data for item two!" << std::endl;
            return -1;
        }
    } else {
        std::cerr << "No data for item one!" << std::endl;
        return -1;
    }
}
```



```
int ex1_22(){
    Sales_data total, d;
    double price = 0;
    if (std::cin >> total.bookNo >> total.units_sold >> price) {
        total.revenue = total.units_sold * price;
        while (std::cin >> d.bookNo >> d.units_sold >> price) {
            d.revenue = d.units_sold * price;
            if (total.bookNo == d.bookNo) {
                total.units_sold += d.units_sold;
                total.revenue += d.revenue;
            } else {
                std::cout << total.bookNo << " " << total.units_sold << " " << total.revenue << " ";
                if (total.units_sold != 0)
                    std::cout << total.revenue / total.units_sold << std::endl;
                else
                    std::cout << "(no sales)" << std::endl;
                total.bookNo = d.bookNo;
                total.units_sold = d.units_sold;
                total.revenue = d.revenue;
            }
        }
        std::cout << total.bookNo << " " << total.units_sold << " " << total.revenue << " ";
        if (total.units_sold != 0)
            std::cout << total.revenue / total.units_sold << std::endl;
        else
            std::cout << "(no sales)" << std::endl;
    } else {
        std::cerr << "No data for item!" << std::endl;
        return -1;
    }
    return 0;
}
```



uOttawa

Ahmedou Jreivine

```

int ex1_23(){
    Sales_data total, d;
    double price = 0;
    if (std::cin >> total.bookNo >> total.units_sold >> price) {
        int cnt = 1;
        while (std::cin >> d.bookNo >> d.units_sold >> price) {
            if (total.bookNo == d.bookNo) {
                ++cnt;
            } else {
                std::cout << total.bookNo << " " << cnt << std::endl;
                total.bookNo = d.bookNo;
                total.units_sold = d.units_sold;
                cnt = 1;
            }
        }
        std::cout << total.bookNo << " " << cnt << std::endl;
    } else {
        std::cerr << "No data for item!" << std::endl;
        return -1;
    }
    return 0;
}

int main() {
    //ex1_20();
    //ex1_21();
    //ex1_22(); // Also for ex1.25
    //ex1_23(); // Also for ex1.24
    return 0;
}

```



uOttawa

Ahmedou Jreivine

## Exercise 7.2:



- Add the ***combine*** and ***isbn*** members to the Sales\_data class you wrote for the exercises in § 2.6.2 (p. 76).

## Solution 7.2:

```
#include <string>
#include <iostream>
struct Sales_data {
    std::string isbn() const { return bookNo; }
    Sales_data &combine(const Sales_data &);
    std::string bookNo;
    unsigned units_sold = 0;
    double revenue = 0.0;
};
Sales_data &Sales_data::combine(const Sales_data &rhs) {
    units_sold += rhs.units_sold;
    revenue += rhs.revenue;
    return *this;
}
int main() {
    return 0;
}
```



## Exercise 7.3:

- Revise your transaction-processing program from § 7.1.1 (p. 256) to use these members.

```
#include <string>
#include <iostream>
struct Sales_data {
    std::string isbn() const { return bookNo; }
    Sales_data &combine(const Sales_data &);
    std::string bookNo;
    unsigned units_sold = 0;
    double revenue = 0.0;
};
Sales_data &Sales_data::combine(const Sales_data &rhs) {
    units_sold += rhs.units_sold;
    revenue += rhs.revenue;
    return *this;
}
```



```

int main() {
    Sales_data total;
    double price;
    if (std::cin >> total.bookNo >> total.units_sold >> price) {
        total.revenue = total.units_sold * price;
        Sales_data trans;
        while (std::cin >> trans.bookNo >> trans.units_sold >> price) {
            trans.revenue = trans.units_sold * price;
            if (total.isbn() == trans.isbn()) {
                //if (total.bookNo == trans.bookNo) {
                    total.combine(trans);
                    //total.units_sold += trans.units_sold;    //total.revenue += trans.revenue;
            } else {
                cout << total.bookNo << " " << total.units_sold << " " << total.revenue << endl;
                total = trans; // Use default copy constructor
                //total.bookNo = trans.bookNo;
                //total.units_sold = trans.units_sold; //total.revenue = trans.revenue;
            }
        }
        std::cout << total.bookNo << " " << total.units_sold << " " << total.revenue << std::endl;
    } else {
        std::cerr << "No data!" << std::endl;
        return -1;
    }
}
return 0;
}

```



uOttawa

Ahmedou Jreivine

# Midterm Exam PART A



## 1. Given the following declaration [1]

```
bool findNumber( int array[], int arraySize, int num2Find );  
call the function findNumber given the variables  
int array[]={ 7, -4, 128, -3199, -5 };  
int num2Find = 42;  
bool result;
```

**result=findNumber(array, 5, 42);**

# Midterm Exam PART A



## 2. What is printed by the following? [1]

```
int tab[] = { 1, 3, 5, 7, 9 };
int *p{tab};
*(p + 2) = 8;
p += 4;
*p = 2;
*(p - 3) = 6;
*(--p) = 0;
for (auto x : tab) {
    std::cout << x << " ";
}
• 1 6 8 0 2
```

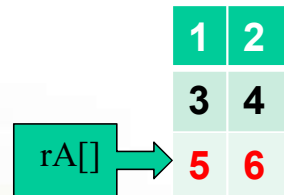
# Midterm Exam PART A



## 3. What is printed by the following? [1]

```
int array[][2]{ 1, 2, 3, 4, 5, 6 };  
int (&rA)[2] = array[2];  
std::cout << rA[1];
```

**// 6 is printed**



# Midterm Exam PART A



## 4. What is printed by the following? [1]

```
int i = 3;
int foo() {
    static int i = 7;
    return ++i;
}
int main() {
    int i = 2;
    std::cout << i << std::endl;
    return i;
}
```

**// 2 is printed ( the local variable is accessed first)**



## Midterm Exam PART A

### 5. What is printed by the following? [1]

```
void foo( int a, int& b, int* c ) {  
    *c = b = a;  
    c++;  
    b *= 2;  
}  
  
int main() {  
    int u=1, v=2, w=3;  
    foo( u, v, &w );  
    std::cout << u << v << w << std::endl;  
    return 0;  
}
```



**// 121 is printed (v and w are affected)**  
**uOttawa**

Ahmedou Jreivine

## Midterm Exam PART A



### 6. What is printed by the following? [1]

```
int *pt[2];  
int t1[] = { 1,2,3 };  
int t2[] = { 9,8,7 };  
pt[1] = t2;  
*pt = t1;  
std::cout << **pt << std::endl;
```

**// 1 is printed**



## Midterm Exam PART B

1.

Mark the lines where a delete statement is in error or is missing and correct them

```
void newArray( int **pArray, int sz ) {  
    if ( pArray != nullptr ) {  
        *pArray = new int[sz];  
    }  
    return;  
}  
int main() {  
    int A[] { 1, 2, 3 };  
    int* pB = &A[1];  
    int* pC = nullptr;  
    int** pD = new int*;  
    newArray(&pC, 4);  
    newArray(pD, 3);  
    // delete[] pB; // Error pB points to auto array A  
    delete[] pC; // pC was allocated with new  
    delete[] *pD; // *pD points to dynamic array  
    delete pD; // pD is a single ptr to a ptr  
    return 0;  
}
```



uOttawa

Ahmedou Jreivine



## PART C:

Consider the following main program:

```
int main() {  
    double a[][3]{{ 2.0, -3.0, 2.0 },  
                  { -4.0, 4.0, 1.0 }};  
    double mVal = maxValue( &a[0][0], 2, 3 );  
    cout << mVal << endl << endl;  
    print( a );  
    double* pVal = minValue( a[0] );  
    cout << endl << *pVal << endl;  
    return 0;  
}
```

Program Output:

```
4  
2 -3 2  
-4 4 1  
-3
```

## function max**Value** in 2Darray.

```
double maxValue(double *ds, int rows, int cols) {  
    double current = *ds;  
    for (int i = 1; i < rows*cols; i++)  
        if (ds[i] > current)  
            current = ds[i];  
    return current;  
}
```

## function print printing the 2D array



```
void print(double (&dr)[2][3]) {  
    for ( auto& row : dr ) {  
        for ( auto col : row ) {  
            std::cout << col << " ";  
        }  
        std::cout << std::endl;  
    }  
}
```

## function minValue in 2Darray.

```
double *minValue(double ds[], int size) {  
    double *current = &ds[0];  
    for (int i = 1; i < size; i++)  
        if (ds[i] < *current)  
            current = &ds[i];  
    return current;  
}
```

2.

Consider the following definition of the function filterInPlace and its main routine:

```
int* filterInPlace( int* beginA, int* endA, bool testInt(int)) {
    int* destA = beginA;
    while( beginA < endA ) {
        if ( testInt( *beginA ) {
            *destA = *beginA;
            ++destA
        }
        ++beginA;
    }
    return destA;
}

int main() {
    int iArray[] { 1, 3, 6 , 7 , 1, -6, -4, -2 };
    int* last = filterInPlace( begin(iArray), end(iArray), greater2 );
    for (auto v : iArray ) {
        std::cout << v << " ";
    }
    cout << endl;
    return 0;
}
```

**//3 6 7 7 1 -6 -4 -2 are printed**



uOttawa

Ahmedou Jreivine

# Refereces



## Accreditation:

- This presentation is prepared/extracted from the following resources:
  - C++ Primer, Fifth Edition.  
Stanley B. Lippman Josée Lajoie Barbara E. Moo
  - <https://github.com/jaege/Cpp-Primer-5th-Exercises>
  - <https://github.com/Mooophy/Cpp-Primer>