

lesson 37 CPP Static Member

شركة **Gammal Tech** تمتلك حساب بنكي واحد، لكن هناك عدد كبير من الموظفين في الشركة يستطيعون استخدام نفس الحساب. على سبيل المثال، قسم المبيعات عند اشتراك أي فرد جديد في الموقع، سواء اشتراك شهري أو سنوي يقوم القسم بتحويل قيمة هذه الاشتراكات إلى حساب البنك الخاص بالشركة، وهناك بعض الموظفين مهمتهم دفع الفواتير كالكهرباء أو التليفونات أو السيرفرات وهذه الأموال ستكون من نفس حساب البنك، وهناك قسم الحسابات وهؤلاء وظيفتهم عمل موازنة بين الاشتراكات والمبالغ المسحوبة (الفواتير)، هل المبالغ التي تدخل الحساب أكبر أم أصغر من المبالغ التي تدفع في الفواتير حتى لا يحدث مشكلة هو حساب واحد، لكن الكل يتعامل معه، لكن كل قسم له صلاحيات محددة دعونا نكتب برنامج لهذا الأمر:

```
#include <iostream>
```

```
using namespace std;
```

```
/*
```

```
float: 4 byte يقوم بتخزين
```

```
double: 8 byte يقوم بتخزين
```

```
يقوموا بتخزين عدد عشري و double يقوم بحجز أرقام أكبر
```

```
*/
```

```
class GT_Bank_Account {
```

```
public:
```

```
double balance;
```

// نحجز متغير للرصيد

void deposit(double a) {

// يقوم بالإيداع

balance += a;

}

void withdraw(double a) {

// يقوم بالسحب

balance -= a;

}

void getBalance() {

cout << balance << endl;

}

};

int main() {

GT_Bank_Account sales;

sales.deposit(100);

sales.deposit(350);

// قسم المبيعات

// قاموا بإضافة 100 و 350

GT_Bank_Account bills;

bills.withdraw(50);

// قسم الفواتير قاموا بدفع 50 للفواتير

```
GT_Bank_Account accountant;  
accountant.getBalance();  
// إذن يجب إن يكون هناك الآن 400  
}
```

output:

0

لكن لماذا قام بطباعة 0 لأننا قمنا بعمل نسخ من class و كل نسخة ليس لها علاقة بالأخرى ولحل هذه المشكلة يجب علينا استخدام static variable وهو متغير يستطيع كل object التعامل معه وإذا تغير قيمته في object، تكون هذه هي القيمة له في object الآخر وسيكون كالتالي:

```
#include <iostream>  
using namespace std;  
  
class GT_Bank_Account {  
public:  
    static double balance;  
    // نحجز متغير للرصيد  
  
    void deposit(double a) {  
        // يقوم بالإيداع  
        balance += a;  
    }  
  
    void withdraw(double a) {  
        // يقوم بالسحب  
        balance -= a;  
    }  
}
```

```

}
void getBalance() {
    cout << balance << endl;
}

};

```

```
double GT_Bank_Account::balance=0;
```

// بهذه الطريقة يستطيع الكل التعامل معها

```
int main() {
    GT_Bank_Account sales;
    sales.deposit(100);
    sales.deposit(350);
    // قسم المبيعات
    // قاموا بإضافة 100 و 350

```

```

    GT_Bank_Account bills;
    bills.withdraw(50);
    // قسم الفواتير قاموا بدفع 50 للفواتير

```

```

    GT_Bank_Account accountant;
    accountant.getBalance();
}

```

output:

400

لكن هناك مشكلة وهى أن كل object تستطيع التعامل مع الحساب سواء الإيداع أو السحب أو معرفة الرصيد
لكن سنقوم بإعطاء كل قسم صلاحيات محددة وسيكون كالتالى :

```
#include <iostream>
```

```
using namespace std;
```

```
class GT_Bank_Account {
```

```
public:
```

```
    static double balance;
```

```
    // نحجز متغير للرصيد
```

```
};
```

```
class sales : public GT_Bank_Account {
```

```
public:
```

```
    void deposit(double a) {
```

```
        // يقوم بالإيداع
```

```
        balance += a;
```

```
    }
```

```
    // قمنا بتحديد صلاحيات قسم المبيعات وعمل ميراث له من class الأساسى
```

```
};
```

```
class bills :public GT_Bank_Account {
```

```
public:
```

```
    void withdraw(double a) {
```

```
        // يقوم بالسحب
```

```
        balance -= a;
```

```
    }
```

```
    // قمنا بتحديد صلاحيات قسم الفواتير وعمل ميراث له من class الأساسى
```

```
};
```

```
class accountant :public GT_Bank_Account {
```

```
public:
```

```
void getBalance() {
```

```
    cout << "balance: " << balance << endl;
```

```
}
```

```
// قمنا بتحديد صلاحيات قسم الحسابات وعمل ميراث له من class الأساسي
```

```
};
```

```
double GT_Bank_Account::balance = 0;
```

```
// بهذه الطريقة يستطيع الكل التعامل معها
```

```
int main() {
```

```
    sales s;
```

```
    s.deposit(100);
```

```
    s.deposit(3500);
```

```
    // للإيداع
```

```
    bills b;
```

```
    b.withdraw(50);
```

```
    // للسحب
```

```
    accountant a;
```

```
    a.getBalance();
```

```
    // طباعة الرصيد
```

```
}
```

```
output:
```

balance: 3550