

CÁC ĐẶC ĐIỂM MỚI CỦA C++ (T.T)

C++



Microsoft®

Visual Studio®

NỘI DUNG

1

Một số lưu ý

2

Ngôn ngữ C++

3

Tham số mặc nhiên

4

Tái định nghĩa hàm

5

Quản lý bộ nhớ động

6

Truyền tham số

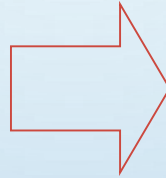
7

Inline Functions

TÁI ĐỊNH NGHĨA HÀM

❖FUNCTIONS OVERLOADING

```
int abs(int i);  
long labs(long l);  
double fabs(double d);
```



```
int abs(int i);  
long abs(long l);  
double abs(double d);
```

❖C++ CHO PHÉP ĐỊNH NGHĨA CÁC HÀM TRÙNG TÊN.

TÁI ĐỊNH NGHĨA HÀM

❖ QUI TẮC TÁI ĐỊNH NGHĨA:

- CÁC HÀM **TRÙNG TÊN** PHẢI **KHÁC** NHAU VỀ **THAM SỐ**:
 - SỐ LƯỢNG
 - THỨ TỰ
 - KIỂU DỮ LIỆU

❖ QUI TẮC GỌI HÀM?

- TÌM HÀM CÓ KIỂU THAM SỐ PHÙ HỢP
- DÙNG PHÉP ÉP KIỂU TỰ ĐỘNG
- TÌM HÀM GẦN ĐÚNG (PHÙ HỢP) NHẤT

TÁI ĐỊNH NGHĨA HÀM

❖ Ví dụ 1:

```
int    Max (int a, int b)
    { return (a>b) ? a : b; }
float  Max (float a, float b)
    { return (a>b) ? a : b; }
SinhVien  Max (SinhVien a, SinhVien b)
    {return  (a.diemtb > b.diemtb) ? a : b;  }
void main() {
    int    x1=1, y1=2;
    float  x2=3, y2=4;
    long   x3=5, y3=6;
    cout << Max(x1,y1)<<"\t"<<Max(x2,y2)<<endl;
    cout << Max(x3,y1) <<endl;
    cout << Max(x3,y2) <<endl;
    cout << Max(x3,y3) <<endl;
}
```

TÁI ĐỊNH NGHĨA HÀM

❖ Ví dụ 2:

```
int      F (int a=0, int b=1)
    { ... }
float    F (float a=5, float b=9)
    { ... }
void main() {
    int      x1=1, y1=2;
    float    x2=3, y2=4;
    long     x3=5, y3=6;
    cout << F(x1) << "\t" << F(y2) << endl;
    cout << F(x3) << F() << endl;
}
```

TOÁN TỬ QUẢN LÝ BỘ NHỚ ĐỘNG

❖ TOÁN TỬ CẤP PHÁT BỘ NHỚ ĐỘNG **NEW**

INT *X;

X = **NEW INT**; //X = (INT*)MALLOC(SIZEOF(INT));

CHAR *Y;

Y = **NEW CHAR**[100]; //Y = (CHAR*)MALLOC(100);

❖ TOÁN TỬ GIẢI PHÓNG VÙNG NHỚ ĐỘNG **DELETE**

DELETE X; // FREE(X);

DELETE []Y; // FREE(Y);

TRUYỀN THAM SỐ

❖ TRUYỀN THEO GIÁ TRỊ (THAM TRỊ)

- GIÁ TRỊ THAM SỐ KHI RA KHỎI HÀM SẼ KHÔNG THAY ĐỔI.

❖ TRUYỀN THEO ĐỊA CHỈ (THAM CHIẾU)

- GIÁ TRỊ THAM SỐ KHI RA KHỎI HÀM CÓ THỂ THAY ĐỔI.

THAM CHIẾU

- ❖ THAM CHIẾU LÀ ĐỊA CHỈ VÙNG NHỚ ĐƯỢC CẤP PHÁT CHO MỘT BIẾN.
- ❖ KÝ HIỆU & ĐẶT TRƯỚC BIẾN HOẶC HÀM ĐỂ XÁC ĐỊNH THAM CHIẾU CỦA CHÚNG
- ❖ VÍ DỤ 1:
 - `INT X = 10, *PX = &X, &Y = X;`
 - `*PX = 20;`
 - `Y = 30;`

THAM CHIẾU

❖ VÍ DỤ 2:

- `INT ARRGET(INT *A, INT I) { RETURN A[I]; }`
- `ARRGET(A, 1) = 1; // A[1] = 1;`
- `CIN >> ARRGET(A,1); // CIN >> A[1];`

❖ VÍ DỤ 3:

- `VOID SWAP1(INT X, INT Y) { INT T = X; X = Y; Y = T; }`
- `VOID SWAP2(INT *X, INT *Y) { INT *T = X; X = Y; Y = T; }`
- `VOID SWAP3(INT &X, INT &Y) { INT T = X; X = Y; Y = T; }`

HÀM INLINE

- ❖ HÀM INLINE HAY CÒN GỌI LÀ HÀM NỘI TUYẾN.
- ❖ SỬ DỤNG TỪ KHÓA **INLINE**
- ❖ YÊU CẦU TRÌNH BIÊN DỊCH **COPY CODE VÀO TRONG CHƯƠNG TRÌNH** THAY VÌ THỰC HIỆN LỜI GỌI HÀM:
 - GIẢM THỜI GIAN THỰC THI CHƯƠNG TRÌNH
 - TĂNG KÍCH THƯỚC CỦA MÃ LỆNH THỰC THI
- ❖ **CHỈ NÊN ĐỊNH NGHĨA INLINE KHI HÀM CÓ KÍCH THƯỚC NHỎ**

HÀM INLINE

❖ Ví dụ:

```
inline float sqr(float x) {  
    return (x*x);  
}  
inline int Max(int a, int b) {  
    return ((a>b) ? a : b) ;  
}
```

BÀI TẬP 1

❖ TÌM LỖI SAI CHO CÁC KHAI BÁO PROTOTYPE HÀM DƯỚI ĐÂY (CÁC HÀM NÀY TRONG CÙNG MỘT CHƯƠNG TRÌNH):

`INT FUNC1 (INT);`

`FLOAT FUNC1 (INT);`

`INT FUNC1 (FLOAT);`

`VOID FUNC1 (INT = 0, INT);`

`VOID FUNC2 (INT, INT = 0);`

`VOID FUNC2 (INT);`

`VOID FUNC2 (FLOAT);`

BÀI TẬP 2

❖ CHO BIẾT KẾT XUẤT CỦA CHƯƠNG TRÌNH SAU:

```
VOID FUNC (INT I, INT J = 0 ){  
    COUT << "SO NGUYEN: " << I << " " << J << ENDL;  
}  
VOID FUNC (FLOAT I = 0, FLOAT J = 0){  
    COUT << "SO THUC:" << I << " " << J << ENDL;  
}  
VOID MAIN(){  
    INT I = 1, J = 2;  
    FLOAT F = 1.5, G = 2.5;  
    FUNC();          FUNC(I);  
    FUNC(F);         FUNC(I, J);  
    FUNC(F, G);  
}
```

BÀI TẬP 3

- a. VIẾT CHƯƠNG TRÌNH NHẬP VÀO MỘT PHÂN SỐ, RÚT GỌN PHÂN SỐ VÀ XUẤT KẾT QUẢ.
- b. VIẾT CHƯƠNG TRÌNH NHẬP VÀO HAI PHÂN SỐ, TÌM PHÂN SỐ LỚN NHẤT VÀ XUẤT KẾT QUẢ.
- c. VIẾT CHƯƠNG TRÌNH NHẬP VÀO HAI PHÂN SỐ. TÍNH TỔNG, HIỆU, TÍCH, THƯƠNG GIỮA CHÚNG VÀ XUẤT KẾT QUẢ.