



Lập trình hướng đối tượng

Nội dung



1

Thông tin môn học

2

Đánh giá kết quả

3

Nội dung môn học

4

1 bài tập C

Thông tin môn học



- ❖ Tên: **Lập trình hướng đối tượng - OOP**
(*Object Oriented Programming*)
- ❖ Số tín chỉ: 3
- ❖ Lý thuyết: 30; Thực hành 30

Đánh giá kết quả



- ❖ Bài thi cuối kỳ + Thực hành: **50%**
- ❖ Giữa kỳ: **30%**
- ❖ Chuyên cần: **20%**
- ❖ Ngôn ngữ thực hành: C++ với Dev C++, Borland C++, VS C++

Nội dung môn học



- ❖ Chương 1: Ngôn ngữ lập trình C++
- ❖ Chương 2: Tổng quan về lập trình hướng đối tượng
- ❖ Chương 3: Lớp & đối tượng
- ❖ Chương 4: Overload toán tử và hàm
- ❖ Chương 5: Sự kế thừa
- ❖ Chương 6: Tính đa hình

Lưu ý về phong cách lập trình



- ❖ Đặt tên
- ❖ Khai báo prototype trước main()
- ❖ {}

Problem – Đặt vấn đề



❖ Nhập 4 biến nguyên và xuất các giá trị vừa nhập

Solution – Giải pháp



1. Dùng 4 biến => cách dài nhất, cơ bản nhất
2. Dùng mảng => khai báo biến gọn hơn, 1 lần thay cho nhiều lần
3. Dùng mảng và vòng lặp do while => viết code nhập gọn hơn, viết 1 lần thay cho nhiều lần
4. Dùng mảng và vòng lặp for => viết code gọn hơn, vì vòng for viết gọn hơn vòng while
5. Dùng mảng, vòng lặp for gộp => viết code gọn hơn, nhưng không tách riêng được 2 phần nhập xuất

Giải pháp



6. Dùng hàm (function) để tách riêng phần nhập xuất => code có thể tái sử dụng nhiều lần
7. Dùng file để nhập xuất từ file thay cho việc nhập bằng bàn phím và xuất ra màn hình

Giải pháp



Cách 1:

```
int a1, a2, a3, a4;  
printf("\n a1 = ");  
scanf("%d", &a1);  
printf("\n a2 = ");  
scanf("%d", &a2);  
printf("\n a3 = ");  
scanf("%d", &a3);  
printf("\n a4 = ");  
scanf("%d", &a4);  
printf("\n Ban vua nhap vao 4 so: %d %d %d %d", a1, a2, a3, a4);
```

```
a1 = 3  
a2 = 4  
a3 = 7  
a4 = 9  
Ban vua nhap vao 4 so: 3 4 7 9_
```

Giải pháp



Cách 2:

```
int a[4];  
printf("\n a1 = ");  
scanf("%d", &a[0]);  
printf("\n a2 = ");  
scanf("%d", &a[1]);  
printf("\n a3 = ");  
scanf("%d", &a[2]);  
printf("\n a4 = ");  
scanf("%d", &a[3]);  
  
printf("\n Ban vua nhap vao 4 so: %d %d %d %d", a[0], a[1], a[2], a[3]);
```

Cách 3: dùng vòng lặp do while

```
int a[4], i;  
i=0;  
do  
{  
    printf("\n a%d = ", i);  
    scanf("%d", &a[i]);  
    i++;  
} while (i<4);  
  
i=0;  
do  
{  
    printf("%d ", a[i]);  
    i++;  
} while (i<4);
```

Cách 4: dùng vòng lặp for

```
int a[4], i;  
for (i=0; i<4; i++) {  
    printf("\n a%d = ", i);  
    scanf("%d", &a[i]);  
}  
  
printf("\n Ban vua nhap vao 4 so: ");  
for (i=0; i<4; i++) {  
    printf("%d ", a[i]);  
}
```

Cách 5:

```
int a[4], i;  
for (i=0; i<4; i++) {  
    printf("\n a%d = ", i);  
    scanf("%d", &a[i]);  
    printf("%d ", a[i]);  
}
```

Giải pháp

Cách 6 :

```
void nhap(int []);  
void xuat(int []);
```

```
void main() {  
    int a[4];  
    nhap(a);  
    xuat(a);  
}
```

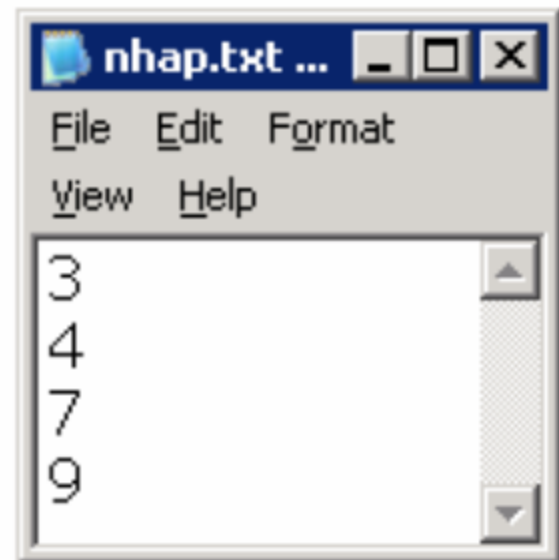
```
void nhap(int b[]) {  
    int i;  
    for (i=0; i<4; i++) {  
        printf("\n a%d = ", i);  
        scanf("%d", &b[i]);  
        printf("%d ", b[i]);  
    }  
}
```

```
void xuat(int c[]) {  
    printf("\n 4 so: ");  
    for (i=0; i<4; i++) {  
        printf("%d ", c[i]);  
    }  
}
```

```

void nhap(int b[], char *f) {
    int i;
    FILE *fp;
    fp = fopen(f, "r");
    for (i=0; i<4; i++) {
        fscanf(fp, "%d", &b[i]);
    }
    fclose(fp);
}

```



```

void xuat(int c[], char *f) {
    int i;
    FILE *fp;
    fp = fopen(f, "w");
    for (i=0; i<4; i++) {
        fprintf(fp, "%d ", c[i]);
    }
    fclose(fp);
}

```

