

Introduction

Bài tập 1

Cho biết trong đoạn mã sau, đối tượng nào có nhiều tên?

```
int *p = new int;
```

```
int *q = p;
```

- ☐ p
- ☐ q
- ☐ Đối tượng được tạo ra bởi new int
- ☐ int

Chưa đúng, p là đối tượng dữ liệu kiểu con trỏ và chỉ có 1 tên là p

Chưa đúng, q là đối tượng dữ liệu kiểu con trỏ và chỉ có 1 tên là q

Đúng

int là một kiểu, không phải là đối tượng dữ liệu

Solution

1. Wrong
2. Wrong
3. Correct Option
4. Wrong

Bài tập 2

Đối tượng dữ liệu có nhiều tên trong đoạn mã C trên có những tên gì?

- ☐ p
- ☐ p và q
- ☐ *p và q
- ☐ *p và *q

Chưa đúng, phải kể ra ít nhất 2 tên của cùng 1 đối tượng dữ liệu

Chưa đúng, p và q là hai tên của hai đối tượng dữ liệu kiểu con trỏ. Một đối tượng dữ liệu kiểu con trỏ có tên là p, đối tượng kia có tên là q.

Chưa đúng, *p là tên của đối tượng được tạo ra bởi lệnh new int, trong khi q là tên của đối tượng kiểu con trỏ.

Đúng, dựa vào 2 tên này, programmer có thể truy xuất (đọc, viết) trên đối tượng được tạo ra bởi new int

Solution

1. Wrong
2. Wrong
3. Wrong
4. Correct Option

Bài tập 3

Cho đoạn mã C++ sau:

```
template <class myType>
myType GetMax (myType a, myType b) {
    return (a>b?a:b);
}
```

Cho biết đoạn mã trên có thể là ví dụ minh họa cho khái niệm nào?

- ☐ Alias - bí danh
- ☐ Polymorphism - Đa hình
- ☐ Garbage-Rác
- ☐ Dangling reference-Tham chiếu treo

Chưa đúng, bí danh là khái niệm về trường hợp cùng một đối tượng dữ liệu có nhiều tên khác nhau. Trong ví dụ này không có đối tượng nào có nhiều tên

Đúng, đây là một ví dụ của đa hình (polymorphism): cũng với 1 tên là GetMax nhưng có thể tham chiếu đến những hàm khác nhau tùy theo kiểu của myType

Chưa đúng, đoạn mã này không tạo ra rác (đối tượng dữ liệu không thể truy xuất)

Chưa đúng, đoạn mã này không tạo ra tham chiếu treo (đối tượng dữ liệu thì mất nhưng tên vẫn còn)

Solution

1. Wrong
 2. Wrong
 3. Correct Option
 4. Wrong
-