**Bài tập 1**

Top of Form

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

Bottom of Form

**Bài tập 2**

Top of Form

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

Bottom of Form

**Bài tập 3**

Top of Form

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 hoạ cho khái niệm nào?



Alias - bí danh



Polymorphism - Đa hình



Garbage-Rác



Dangling reference-Tham chiếu treo

# Exercise 1

Top of Form

Given the following code written in C:

x = a \* b;

What is the binding time of variable x and its type?



Programming Time - Thời gian lập trình



Language Design Time- Thời gian định nghĩa ngôn ngữ



Language Implementation Time - Thời gian hiện thực ngôn ngữ



Runtime - Thời gian chạy

Bottom of Form

# Bài tập 2

Top of Form

Cho x trong đoạn mã của câu trên là biến toàn cục. Cho biết thời gian xảy ra ràng buộc giữa biến x và địa chỉ tuyệt đối của x trong bộ nhớ của máy tính?



Compiling time - Thời gian dịch



Linking time- Thời gian kết nối



Loading time - Thời gian nạp



Runtime - Thời gian chạy

Bottom of Form

# Bài tập 3

Top of Form

Cho biết trên ngôn ngữ C, ràng buộc giữa dấu \* và nghĩa của nó (phép nhân) xảy ra vào thời gian nào?



Language design time - Thời gian định nghĩa ngôn ngữ



Runtime



Language Implementation Time - Thời gian hiện thực ngôn ngữ



Programming time

# Exercise 1

Top of Form

Given the following C fragment:

int x;

void foo(int y) {

    static int z;

    int \* t = malloc(sizeof(int));

    ...

}

Choose the WRONG statement?



x is allocated in static memory



y is allocated in stack memory



z is allocated in static memory



t is allocated in heap memory

Bottom of Form

# Exercise 2

Top of Form

Given the following C fragment:

int x;

void foo(int y) {

    static int z;

    int \* t = malloc(sizeof(int));

    ...

}

Choose the WRONG statement?



The lifetime of x is the same as the lifetime of the whole program



The lifetime of y is the same as the lifetime of function foo



The lifetime of z is the same as the lifetime of function foo



The lifetime of t is the same as the lifetime of function foo

Bottom of Form

# Exercise 3

Top of Form

Given the following C code:

//position 1

int \* foo() {

     // position 2

    x[0] = 1;

    return x;

}

which declaration of x and at which position can cause a runtime error (dangling reference or garbage)?



int x[10]; //position 1



int x[10]; //position 2



static int x[10]; // position 2



int\* x = malloc(10\*sizeof(int)); // position 2

Bottom of Form

Bottom of Form

Bottom of Form