What is the general non-object-oriented sense of the word "object"?

What are the available binding times?

Name three storage allocation mechanisms.

What is a static object?

Name the two kinds of constants.

How can they be allocated?

Name two examples of static objects in C

Static allocation of local variables is impossible in languages with ...

Good compilers will try to keep arguments, return values, and temporaries where?

What is a dynamic link?

Each subroutine has its own ...

What does it hold?

Who maintains the stack?

What is a calling sequence?

What are the prologue and epilogue?

What is the frame pointer?

What is it useful for?

What is displacement addressing?

What is the heap?

It should not be confused with ...

What are the major concerns in heap design?

What is internal fragmentation?

What is external fragmentation?

What is a free list?

How does a free list work?

Name and describe two methods for selecting a heap block for a given object.

Compare the best fit and first fit algorithms.

What is the cost of allocation for memory management algorithms with one free list?

How can memory-allocation be reduced to constant time?

How can such techniques be further subdivided?

What are two common mechanisms for dynamic pool adjustment?

What is the buddy system?

How does the Fibonacci heap work?

It results in ...

An adversary can always devise a series of memory requests that ...

Describe adversarial memory request strategies for static and dynamic allocation schemes.

How can external fragmentation be eliminated?

What are the two main types of manual memory allocation errors?

What is the scope of a binding?

What is lexical scoping?

It is also known as ...

Why is *lexical scope* a better term than *static scope*?

Define dynamic scoping.

What is meant by "scope" when no particular binding is in mind?

What is the referencing environment?

What is the closest nested scope rule?

What is a hole in a scope?

Name two methods languages offer for accessing hidden bindings.

How can subroutines refer to local objects?
To nonlocal ones?

What is a subtlety of static scoping?

When is declare-before-use problematic?

How does C/C++ solve the conundrum?

What are forward references?

What is the consequence of allowing them?

What are the benefits of information hiding?

What do the two uses of the C keyword `static` have in common?

What are modules?