22. Synthesized Member Functions Exercises Solutions

Synthesized+Member+Functions+Exercises.pdf

Synthesized default constructor

What is a default constructor?

✅ Constructor that doesn't take any parameters whether created by the developer in a class or created by compiler when there's no one created.

In what circumstances does the compiler synthesize a default constructor?

✅ When the class doesn't have any constructors defined. ⭐ (including the copy constructor)

What does this synthesized default constructor do?

✅ Initialize the class members. ~~data. (sometimes with garbage value)~~

➕ For class members, it is called their default constructor.

➕ For members of the built-in type, it leaves them uninitialized.

Synthesized copy constructor

In what circumstances does the compiler synthesize a copy constructor?

⭐ When there's no defined copy constructor or in case no constructor is defined at all then it will synthesized all special function members.

✅ the answer is not so bad but it's better to say: "The compiles will synthesize a copy constructor for a class which does not define a copy constructor itself." it doesn't matter if the class has any other constructor or not.

What does this synthesized copy constructor do?

✅ It copies the object into another newly created object.

➕ It copies each member of the class from the corresponding member in the argument object.

➕ For class members, it calls their copy constructor.

➕ For members of the built-in type, it copies them.

Synthesized assignment operator

In what circumstances does the compiler synthesize an assignment operator?

✅ When this is not implemented in a class.

What does this synthesized assignment operator do?

✅ it copies the member data values from an existing object to another existing object. (both objects are the same type).

- It copies each member of the class from the corresponding member in the argument object.
- For class members, it calls their assignment operator.
- For members of the built-in type, it copies them.

Drawbacks of synthesized functions

Give some examples of when allowing the compiler to synthesize special member functions can have undesirable results.

- Built-in types are default initialized and it may be a garbage value.
- Pointer members are “shallow copied”.
- Derived objects are “sliced” on the destructor.

Suggest how to avoid these issues

- Implement default constructor to initialize the class member correctly.
- Implement a copy constructor and assignment operator which makes a “deep copy” of the pointer member.
- Implement an empty destructor in the base class and declare it virtual.

Rule of Three

What is the "Rule of Three"?

- The Rule of Three states that, if any one of the destructor, copy constructor, or assignment operator has to be implemented for a class to work properly, all three should be implemented
- This does not apply to a virtual destructor in a base class which is only there to prevent derived classes being "sliced"

Give an example where the Rule of Three would be helpful

- Class with pointer member which allocates memory in the constructor. This needs a destructor that releases the allocated memory, also a copy constructor and assignment operator to perform a "deep copy". Here, the Rule of Three helps us avoid memory leaks and program crashes.

Synthesized Special Member Functions

Write a program that demonstrates the use of synthesized special member functions. Add comments to your program showing the code that the compiler would generate when synthesizing these member functions

- `synthesized_member_special_fncs.cpp`