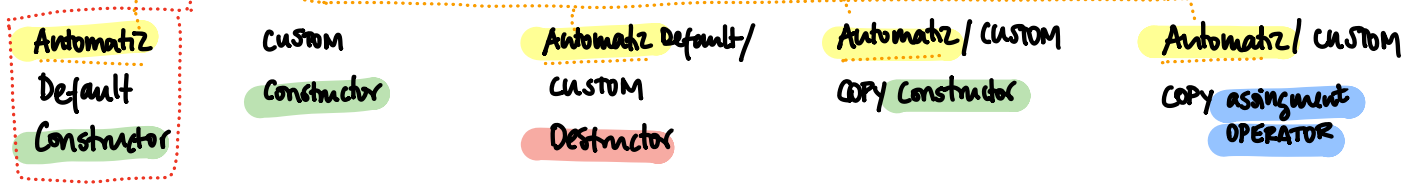
Automatic: if not explicitly done, all these 4 are created for us
BUT: 1. we can overload them any time, in which case they are not automatic
2. The automatic default constructor is **not** created if we explicitly create any other constructor; so don't forget doing it if we create another constructor!



Typical form	ClassName()		~ClassName()		
Examples of usage (when are they called?)	ClassName c;	ClassName c; ClassName C(1);	When stack destroyed	ClassName C1; ClassName C2(C1);	ClassName C1; ClassName C2 = C1;

Implementation examples (if custom)		ClassName::ClassName()
		{ //no need to do anything }
		ClassName::ClassName
		(int i)
		{ m_var = i; }
			ClassName::~ClassName()
			{ // delete if necessary
			// no return }
				ClassName::ClassName
				(const ClassName& rv)
				{ m_var = rv.m_var }
					ClassName&
					ClassName::operator=
					(const ClassName & rv)
					{ m_var = rv.m_var;
					return *this; }

Notes, guidelines	Implement them always if another constructor is implemented!	Note that any time we explicitly define a constructor its implementation must be added too! At least with ∅ initialization	We might want to implement them if some delete is required	We might want to implement it if we want control of which variables are copied & how	We might want to implement it if we want control of which variables are copied & how