Five language features offered by C++ 1985

Ordered by importance with commenting their role in programming.

1. **Classes and Derived Classes** Enables to represent user defined types.
2. **Public / Private access control** Enables information hiding and user interface.
3. **Constructors and Destructors** Enables concise resource handling and class invariant.
4. **Operator / Function Overloading** Simplifies and reduces the amount of code that needs to be written.
5. **Virtual Functions** Enables to define general types without concrete implementation ⇒ abstract classes.

Five language features offered by C++98

Ordered by importance with commenting their role in programming.

1. **RAII** Resource Acquisition Is Initialization ⇒ Copy and Assignment constructors.
2. **Exceptions** Enables to propagate program errors in an elegant way without requiring the need to handle those errors locally.
3. **Templates** Enables class and function objects to be created with type arguments. Requires runtime support.
4. **Namespaces** Enables logical distinction of parts of the program.
5. **STL** Adds support for handling common tasks with containers and algorithms.

Five language features offered by C++11

Ordered by importance by commenting on their role in programming.

1. **Resource Management Pointers** Unique and Shared pointers. Moving objects by pointers from one scope to another and having reference counter pointers without a certain owner.
2. **Move Semantics** Moving objects by reference without copying large objects.
3. **Lambda Expressions** Anonymous function objects (delegates) to support functions that accept delegates.

4. **Concurrency Compliant Memory Model** To support writing applications with multiple threads in a multiprocessor environment.

5. **Type Deduction with `auto`** To reduce the amount of code that needs to be written. Improves readability.