

CPP-05

* Polymorphism

⇒ Allows morphing derived classes into their base class type:

⇒ "is a" vs "has a"

→ Square is a shape: Can inherit from Shape.

→ Car has a wheel: Should not inherit each other.

Google-STYLE

{Prefer composition}

{Include an object of another class
as a member of your class}

Over hierarchies

const Base& base = Derived

Often used for

⇒ Working with all children of some Base class in unified manner.

⇒ Enforcing an interface in multiple class to force them to implement some functionalities.

/

⇒ In Strategic pattern, where some complex functionality is outsourced into separate classes and is passed to the object in a modular fashion.

⇒ function overloading happens at compile time.

& function overriding happens at runtime.

* Using interfaces

⇒ Use interfaces when you must enforce other classes to implement some functionality.

⇒ Allow to easily extend functionality by simply adding a new class.

* Reading and Writing to files

#include <fstream>

→ File Stream

⇒ There are many modes under which a file can be opened.

Mode	Meaning
ios_base::app	append output
ios_base::ate	Seek to EOF when opened
ios_base::binary	open the file in binary mode
ios_base::in	Open the file for reading
ios_base::out	Open the file for writing
ios_base::trunc	Overwrite the existing file

ifstream input ("file.txt", ios_base::in);
↓
name of the object

* String Stream

```
#include <sstream>
```

```
stringstream s_out;
```

* CMake find_path and find_library

→ We can use an external library.

→ Need headers and binary library files.

■ Headers

```
find_path(SOME_PKG_INCLUDE_DIR
```

```
    include/some-file.h  
    <Path1> <Path2> ...)
```

```
include_directories (${SOME_PKG_INCLUDE_DIR})
```

■ Libraries

```
find_library(SOME_LIB
```

```
    NAMES <some-lib>
```

```
    PATHS <Path1> <Path2> ...)
```

```
target_link_libraries(target ${SOME_LIB})
```


* find-package

⇒ find-package calls multiple find-path and find-library functions.
