ComplexShape

```
public:
        std::list<Shape*> shapeList;

         ComplexShape();
         ~ComplexShape() { shapeList.clear() };

         void appendList(ComplexShape& s);
         void addElement(Shape* s);
```

The ‘Application’ object performs operations on the input files and draws the scene.
The ‘Complex Shape’ object utilises polymorphism to draw each of the objects by calling the draw() function to the object pointed to by the ‘Shape’ pointer in the list.
A range of ‘Exception’ classes are defined for use in ‘throw/catch’ exception cases. The scene can be rendered independent of drawing order by parsing the file twice. A warning is displayed if a shape is defined after draw.

See diagram below for program operation flow.

List of Exception Class Types

```
// Exceptions to return:
//  - file it occurred
//  - line it occurred
//  - type of error that occurred

class invalidFileException{};
class invalidDefineHeaderException{};
class invalidDrawHeaderException{};
class invalidDatatypeException{};
class invalidDefineFooterException{};
class invalidDrawFooterException{};
class missingParametersException{};
class outOfSceneException{};
class invalidImageSizeException{};
```

Application

```
private:
        // shape define parse or draw parse
        bool parseType;

        void check_size(istream& file);

        void check_Header(istream& file);
        void check_Footer(istream& file);

        void declareShape(istream& file);

public:
        std::map<string, ComplexShape> shapeMap;

        bool read_file(string filePath);

        void drawScene();
```