

**Domain Specific Language**  
**Shapes to BMP (.stb)**  
**Polymorphism Assignment - 50 points**

Old Example exe: ([download here](#))

Classes to start with: Picture – The type that is used to write the bitmap to a file. [Picture.h](#) [Picture.cpp](#)

RGBTriple - Used from Shape and Picture to represent a color. [RGBTriple.h](#) [RGBTriple.cpp](#)

Classes to implement (At minimum. Feel free to do more.): **[3 points per class]**

- Coordinate – Struct with attributes x & y
- Shape – The abstract base class. All other shapes inherit from Shape.
  - Abstract Methods
    - An initialize method that takes a string as an argument. This string should be the stb syntax that is used to define the shape's position, color, and size. (pure virtual)
    - Draw (pure virtual)
    - DrawShadow (pure virtual)
  - Attributes [ RGBTriple color , Coordinate startCoordinate, bool isFilled (optional)]
- Point
- Line
- RightTriangle
- Star – Can be drawn with crossing lines as you would do by hand. Any number of sides.
- Polygon – to be used for all other irregularly shaped polygons. May have any number of points. Just connect the dots (aka Coordinates) in order. Also connect the last dot to the first dot.
- Circle – you may use an algorithm for this from any source but you must cite the source and write the code yourself.
- Rectangle
- Diamond

Create a main program to: **[20 points for main and stb syntax]**

1. Read a .stb source file that contains the instructions for what shapes to draw.
2. Build a list of shapes `list<Shape*> shapes;` using the shapes instructed in the .stb file. **Dynamically create the shapes so that you can add the addresses to the list.**
3. Create a picture object.
4. Iterate thru the list of shapes and draws each shape to the picture.
5. Create an output file with the same name as the input file but with the file extension .bmp.

STB file requirements:

1. At least one example .stb file should be submitted with your project. Examples of all shapes in your project should be included.
2. The syntax of your language should be intuitive. I should be able to create other images using the syntax I learn from your example.
3. The programmer should be able to set the size and background color of the image.
4. The programmer should be able to set the position, color and size of each shape where appropriate.
5. The programmer should be able to choose if they want the shapes to have a shadow. One indicator in the stb file (all or nothing). All shapes should have a shadow in the same direction. Use a static Boolean to store whether or not to show the shadow and static integers to store the shadow's x and y offset.

Note: An example stb syntax is not provided so that you will be creative and define your own syntax.

Other challenges for fun:

1. Define colors in your .stb file that can be used when defining later shapes.
2. Define reusable shape groupings in your .stb file that can be drawn multiple times. This could be something like a house that is drawn with multiple lines.
3. Allow the user to define if each shape is filled with its color.