### Homework 2: UML Class Diagrams

You may use a UML editor or hand-draw diagrams for the following two problems.

1. Draw a UML Class diagram describing the following C++ classes and their relationships.

* Use the UML stereotypes interface and abstract where appropriate.
* Where appropriate distinguish between class and instance attributes and between class and instance methods.
* Show the visibility and signatures of methods.
* Show the visibility and types of attributes.
* When the type of an attribute is another class, show it using the aggregation/composition relationship instead of showing it inside the attribute compartment.
* Show constructors and destructors.

class IShape {

public:

virtual float area() = 0;

virtual float circumference() = 0;

virtual void rotate() = 0;

virtual void playSound() = 0;

virtual ~IShape() { };

};

class Shape: IShape {

private:

Point center;

float extent; // distance from center to outermost point

std::string soundFileName;

static int rotationDegrees;

public:

void rotate();

void playSound();

static void setRotationDegrees(int);

~Shape() { };

};

class Circle: Shape {

public:

float area();

float circumference();

Circle(Point, float, std::string);

~Circle();

};

class Square: Shape {

private:

float width;

public:

float area();

float circumference();

Square(Point, float, std::string);

~Square();

};

class Point {

public: // yes, this breaks the recommended conventions

float x;

float y;

}

2. Draw a UML class diagram that describes the following things. Use the standard UML stereotypes where appropriate. Show visibility, but do not show method signatures, attribute types, constructors, or destructors. Where appropriate distinguish between class and instance attributes and between class and instance methods. (Assume that all are instance attributes/methods unless otherwise indicated.) When an attribute is not another class in the diagram, show it inside the appropriate compartment. Otherwise show it as an aggregation relationship, with the appropriate role name. Remember to show multiplicities, and no class should appear more than once in the diagram.

* LibraryCard is a class that inherits from IDCard, has public method getID, has protected attribute expirationDate.
* Card is an interface that has public method getName, has public method getID.
* Class Name has public attribute firstName, has public attribute lastName.
* IDCard is a class that implements Card, has protected attribute name (of type Name), has protected attribute id, has public class attribute nextAvailableID, has public method getName, has public method setName, has public method getID, has protected method setID. Show that IDCard contains a Name object.
* BankCard is a class that implements Card, has private attribute name (of type Name), has private attribute id, has private attribute balance, has public method getName, has public method setName, has public method getID, has public method setID, has public method updateBalance. Show that BankCard contains a Name object.