**44-542 Object Oriented Programming**

**UML**

1. Represent the class **Rectangle** using UML notation.

public class Rectangle

{

private double width;

private double height;

public Rectangle(double width, double height)

{

this.width = width;

this.height = height;

}

public Rectangle()

{

width = 0.0;

height = 0.0;

}

public double getHeight()

{

return height;

}

public void setHeight(double height)

{

this.height = height;

}

public double getWidth()

{

return width;

}

public void setWidth(double width)

{

this.width = width;

}

public double getArea()

{

return width \* height;

}

public double getPerimeter()

{

return 2.0 \* (width + height);

}

}

2. Represent the class **Movie** using UML notation.

public class Movie

{

private String title;

private int lengthInMinutes;

private double cost;

private double rentalPrice;

private int timesRented;

public Movie(String title, int lengthInMinutes,

double cost, double rentalPrice)

{

this.title = title;

this.lengthInMinutes = lengthInMinutes;

this.cost = cost;

this.rentalPrice = rentalPrice;

this.timesRented = 0;

}

public String getTitle()

{

return title;

}

public int getTimesRented()

{

return timesRented;

}

public void incrementTimesRented()

{

++timesRented;

}

public boolean isLong()

{

return lengthInMinutes >= 120;

}

public boolean hasMadeProfit()

{

return timesRented \* rentalPrice > cost;

}

public double getProfitAmount()

{

if(hasMadeProfit())

{

return timesRented \* rentalPrice - cost;

}

else

{

return 0.0;

}

}

}