

# COW - Classes & Objects

## Level 1

**Create a class called MyDate that has the following attributes:**

### Variables

- int day – the day
- int month – the month
- int year – the year

### Methods

Name: MyDate (constructor)

Input: int day, int month, int year

Output: none

Action: takes in and sets the initial variable values

Name: getDay

Input: none

Output: int day

Action: returns the day

Name: getMonth

Input: none

Output: int month

Action: returns the month

Name: getYear

Input: none

Output: int year

Action: returns the year

**Create a class called Name that has the following attributes:**

### **Variables**

- String firstName – first name
- String middleName – middle name
- String lastName – last name

### **Methods**

Name: Name (constructor)

Input: String firstName, String middleName, String lastName

Output: none

Action: takes in and sets the initial variable values

Name: getFirstName

Input: none

Output: String firstName

Action: returns the first name

Name: getMiddleName

Input: none

Output: String middleName

Action: returns the middle name

Name: getLastName

Input: none

Output: String lastName

Action: returns the last name

**Create a class called Point that has the following attributes:**

### **Variables**

- int x, y – integer values that determine the location
- Color theColor – Color that determines the Color of the point
- int radius – radius that determines the size

### **Methods**

Name: Point

Input: int x, int y, Color theColor, int size

Output: nothing

Action: a constructor that takes in the initial values of the class variables

Name: getX

Input: nothing

Output: int x

Action: returns the x value

Name: getY

Input: nothing

Output: int y

Action: returns the y value

Name: getColor

Input: nothing

Output: Color theColor

Action: returns theColor

Name: getRadius

Input: nothing

Output: int radius

Action: returns the radius value

Name: paint

Input: Graphics

Output: nothing

Action: draws the point to the screen using the following code:

```
g.setColor(theColor);
```

```
g.drawOval(x-radius, y-radius, radius*2, radius*2);
```

**Create a class called Box that has the following attributes:**

### Variables

- int x, y – integer values that determine the location
- Color theColor – Color that determines the Color of the point
- int width, height – integer values that determine how wide and tall the Box is

### Methods

Name: Box  
Input: int x, int y, Color theColor, int width, int height  
Output: nothing  
Action: a constructor that takes in the initial values of the class variables

Name: getX  
Input: nothing  
Output: int x  
Action: returns the x value

Name: getY  
Input: nothing  
Output: int y  
Action: returns the y value

Name: getColor  
Input: nothing  
Output: Color theColor  
Action: returns theColor

Name: getWidth  
Input: nothing  
Output: int width  
Action: returns the width value

Name: getHeight  
Input: nothing  
Output: int height  
Action: returns the height value

Name: paint  
Input: Graphics  
Output: nothing  
Action: draws the box to the screen using the following code:  
`g.setColor(theColor);`  
`g.fillRect(x, y, width, height);`

**Create a class called Digit that has the following attributes:**

**Variables**

- int value – stores the numerical value of the digit

**Methods**

Name: Digit

Input: int value

Output: nothing

Action: a constructor that takes in the initial values of the class variables

Name: getValue

Input: nothing

Output: int value

Action: returns the value of the digit

## Level 2

### Add the following methods to the MyDate class:

Name: toString

Input: none

Output: String description

Action: returns the date in the form “month \ day \ year”, ex: “1\13\1978”. Hint – use “\\” to insert one slash into a String

### Add the following methods to the Name class:

Name: toString

Input: none

Output: String

Action: returns the first, middle, and last name together in one String with a space separating them

### Add the following methods to the Point class:

Name: moveRight

Input: int amount

Output: nothing

Action: moves the point to the right by the specified amount

Name: moveLeft

Input: int amount

Output: nothing

Action: moves the point to the left by the specified amount

Name: moveUp

Input: int amount

Output: nothing

Action: moves the point up by the specified amount

Name: moveDown

Input: int amount

Output: nothing

Action: moves the point down by the specified amount

Name: moveTo

Input: int nextX, int nextY

Output: nothing

Action: changes the coordinate variables to nextX and nextY

**Add the following methods to the Box class:**

Name: moveRight  
Input: int amount  
Output: nothing  
Action: moves the point to the right by the specified amount

Name: moveLeft  
Input: int amount  
Output: nothing  
Action: moves the point to the left by the specified amount

Name: moveUp  
Input: int amount  
Output: nothing  
Action: moves the point up by the specified amount

Name: moveDown  
Input: int amount  
Output: nothing  
Action: moves the point down by the specified amount

Name: stretchRight  
Input: int amount  
Output: nothing  
Action: stretches the box by the given amount. This should be done by adding amount to the appropriate variable

Name: stretchDown  
Input: int amount  
Output: nothing  
Action: stretches the box by the given amount. This should be done by adding amount to the appropriate variable

Name: moveTo  
Input: int nextX, int nextY  
Output: nothing  
Action: changes the coordinate variables to nextX and nextY

**Add the following methods to the Digit class:**

Name: increment

Input: nothing

Output: nothing

Action: adds one to the digit's value. If the value exceeds nine, then the value should be set to zero

Name: decrement

Input: nothing

Output: nothing

Action: subtracts one to the digit's value. If the value falls below zero, then the value should be set to nine

Name: setValue

Input: int nextValue

Output: nothing

Action: sets value to nextValue



### Level 3

#### Add the following methods to the **MyDate** class:

Name: equals

Input: MyDate otherDate

Output: boolean same

Action: returns whether the date passed in is that same as this date

Name: compareTo

Input: MyDate otherDate

Output: int comparison

Action: the value 0 is returned if the date passed is equal to this date; a value less than 0 if this date comes before the other date; and a value greater than 0 if this date comes after the other date.

#### Add the following methods to the **Name** class:

Name: equals

Input: Name otherName

Output: boolean same

Action: returns whether the name passed in is that same as this name (ie – all the names match up)

Name: compareTo

Input: Name otherName

Output: int comparison

Action: the value 0 is returned if the name passed in equal to this name; a value less than 0 if this name is lexicographically less than the other name; and a value greater than 0 if this string is lexicographically greater than the other name. To compare names, first the last names are compared, then first, then middle.

#### Add the following methods to the **Point** class:

Name: equals

Input: Point otherPoint

Output: boolean same

Action: returns whether the point passed in has the same coordinate (x, y) values as the point passed in.

Name: compareTo

Input: Point otherPoint

Output: int comparison

Action: the value 0 is returned if the point passed in equidistant to the origin (0, 0); a value less than 0 if this point is closer to the origin; and a value greater than 0 if this point is further away from the origin. Hint – the Pythagorean theorem is very useful here.

**Add the following methods to the Box class:**

Name: equals

Input: Box otherBox

Output: boolean same

Action: returns whether other box has the same dimensions (width and height) as this box.

Name: compareTo

Input: Box otherBox

Output: int comparison

Action: the value 0 is returned if the box has the same area as this box; a value less than 0 if this box has a smaller area than the box passed in; and a value greater than 0 if this box has a larger area than the box passed in.

**Add the following methods to the Digit class:**

Name: equals

Input: Digit otherDigit

Output: boolean same

Action: returns whether the digit passed in has the same value

Name: compareTo

Input: Digit otherDigit

Output: int comparison

Action: the value 0 is returned if the digit passed in equal to this digit's value; a value less than 0 if this digit's value less than the other digit's value; and a value greater than 0 if this digit's value is greater than the other digit's value.

## Level 4

**Create a class called Person that has the following attributes:**

### Variables

- Name theName – stores the person's name
- MyDate birthday – stores the person's birthday
- int socialSecurityNumber – stores the person's social security number

### Methods

Name: Person (constructor)

Input: Name theName, MyDate birthday, int socialSecurityNumber

Output: none

Action: takes in and sets the initial variable values

Name: getName

Input: none

Output: Name

Action: returns the name

Name: getBirthDay

Input: none

Output: MyDate

Action: returns the birthday

Name: getSocSecNumber

Input: none

Output: int

Action: returns the socialSecurityNumber

Name: toString

Input: none

Output: String

Action: returns a String description of the person that includes all information about them

Name: equals

Input: Person other

Output: boolean

Action: returns whether the person passed in is the same by comparing social security numbers

Name: compareTo

Input: Person other

Output: integer

Action: Compares this person with person passed in by using their names.

**Create a class called Line class that has the following attributes:**

**Variables**

- Point endPoint1, endPoint2 – the two endpoints of the line

**Methods**

Name: Line

Input: Point endPoint1, Point endPoint2

Output: nothing

Action: a constructor that takes in the initial values of the class variables

Name: getEndPoint1

Input: nothing

Output: Point endPoint1

Action: returns the first endpoint

Name: getEndPoint2

Input: nothing

Output: Point endPoint2

Action: returns the second endpoint

Name: getDistance

Input: nothing

Output: double distance

Action: returns the distance between the endpoints

Name: getMidPoint

Input: nothing

Output: Point midPoint

Action: returns the midpoint of the line. When calculating the midpoint, use integer division and round down.

Name: moveRight

Input: int amount

Output: nothing

Action: moves the line to the right by the specified amount. This should be done by moving each of the two endpoints to the right.

Name: moveLeft

Input: int amount

Output: nothing

Action: moves the line to the left by the specified amount. This should be done by moving each of the two endpoints to the left.

Name: moveUp

Input: int amount

Output: nothing

Action: moves the line up by the specified amount. This should be done by moving each of the two endpoints up.

Name: moveDown

Input: int amount

Output: nothing

Action: moves the line down by the specified amount. This should be done by moving each of the two endpoints down.

Name: equals

Input: Line otherLine

Output: boolean

Action: returns if the other line has the same endpoints. Keep in mind that the endpoints that are the same pair but in a reversed order are still the same line.

Name: compareTo

Input: Point otherPoint

Output: int comparison

Action: the value 0 is returned if the distances of the two lines are the same; a value less than 0 if this line's distance is less than the other line's distance; and a value greater than 0 if this line's distance is greater than the other line's distance.

Name: paint

Input: Graphics

Output: nothing

Action: draws the line to the screen by telling the two endpoints to the screen and a black line between the two

## Level 5

**Create a class called Address that has the following attributes:**

### Variables

- int number – stores the number
- String street – name of the street
- String county – stores the county
- String state – stores the state
- int zipCode – stores the zip code as a 5 digit number
- String country – stores the country

### Methods

Name: Address (constructor)

Input: int number, String street, boolean isApartment, int apartmentNumber, String county, String state, int zipCode, String country

Output: none

Action: takes in and sets the initial variable values

Name: Address (constructor)

Input: int number, String street, String county, String state, int zipCode, String country

Output: none

Action: takes in and sets the initial variable values, this constructor also does not take in apartment information and so isApartment should be set to false and apartmentNumber should be set to 1

Name: getNumber

Input: none

Output: integer

Action: returns the number

Name: getStreet

Input: none

Output: String

Action: returns the street

Name: getCounty

Input: none

Output: String

Action: returns the county

Name: getState

Input: none

Output: String

Action: returns the state

Name: getZipCode

Input: none

Output: int

Action: returns the zip code

Name: `getCountry`  
Input: `none`  
Output: `String`  
Action: returns the country

Name: `toString`  
Input: `none`  
Output: `String`  
Action: returns a `String` description of the address to appear in the following format:  
                    "1234 Milky Way Lane  
                    Sterling, VA 20165  
                    USA"

Hint – use “`\n`” to put in a carriage return

Name: `equals`  
Input: `Address other`  
Output: `boolean`  
Action: returns whether the address object passed in is the same

**Add the following methods to the Box class:**

### **Variables**

- boolean isOn

### **Methods**

Change the constructor(s) to the following:

Name: Box

Input: int x, int y, Color theColor, int width, int height

Output: nothing

Action: a constructor that takes in the initial values of the class variables. It also sets isOn to false.

Add the following methods

Name: getIsOn

Input: nothing

Output: boolean isOn

Action: returns isOn

Name: turnOn

Input: nothing

Output: nothing

Action: sets isOn to true

Name: turnOff

Input: nothing

Output: nothing

Action: sets isOn to false

Change the following method

Name: paint

Input: Graphics

Output: nothing

Action: draws the box to the screen by setting the color and using fillRect. If isOn is false then the color should be set to black (Color.BLACK) instead of theColor.



**Add the following methods to the Digit class:**

**Variables**

- Box topBar, middleBar, bottomBar, upperRightBar, lowerRightBar, upperLeftBar, lowerLeftBar  
– the Boxes that will be used to display the digit to the screen

**Methods**

Change the constructor(s) to the following:

Name: Digit

Input: int value, Box topBar, Box middleBar, Box bottomBar, Box upperRightBar, Box lowerRightBar, Box upperLeftBar, Box lowerLeftBar

Output: nothing

Action: a constructor that takes in the initial values of the class variables and calls updateDisplay

Add the following methods

Name: getBar

Input: String whichBar

Output: Box theBar

Action: returns the Box that corresponds to the name passed in which could be “Top”, “Middle”, “Bottom”, “Upper Right”, “Lower Right”, “Upper Left”, “Lower Left”

Name: paint

Input: Graphics g

Output: nothing

Action: calls the paint method on each of the Boxes that are stored

Name: updateDisplay

Input: nothing

Output: nothing

Action: turns the boxes on or off so that the correct number is displayed that corresponds to the value stored.

## Level 6

**Add the following methods to the Address class:**

### Variables

- boolean isApartment – whether or not it is an apartment
- int apartmentNumber – the number of the apartment

### Methods

Change the constructor(s) to the following:

Name: Address (constructor)

Input: int number, String street, boolean isApartment, int apartmentNumber, String county, String state, int zipCode, String country

Output: none

Action: takes in and sets the initial variable values

Name: Address (constructor)

Input: int number, String street, String county, String state, int zipCode, String country

Output: none

Action: takes in and sets the initial variable values, this constructor also does not take in apartment information and so isApartment should be set to false and apartmentNumber should be set to 1

Name: isApartment

Input: none

Output: boolean

Action: returns isApartment

Name: getApartmentNumber

Input: none

Output: int

Action: returns the apartment number

Modify the toString method to return a String in the format if the address is an apartment:

```
"986 Glowood Drive, Apt 227  
Pittsburgh, PA 15227  
USA"
```

Modify the equals method to take into account whether the address is an apartment.

## Add the following methods to the Box class:

### Methods

Add the following methods

Name: stretchLeft

Input: int amount

Output: nothing

Action: changes the width of the rectangle so that the left edge of the Box moves left by amount passed in but the right edge stays where it is

Name: stretchUp

Input: int amount

Output: nothing

Action: changes the height of the rectangle so that the top edge of the Box moves up by amount passed in but the bottom edge stays where it is

Name: stretch

Input: int amount

Output: nothing

Action: changes the width and height of the rectangle so that each edge moves further or closer from the center by amount

Change the following method

Name: paint

Input: Graphics

Output: nothing

Action: draws the box to the screen by setting the color and using fillRect. If isOn is false then the color should be set to black (Color.BLACK) instead of theColor. It should also draw a border using drawRect that has a line thickness of 5 and a color of gray (Color.GRAY). The line thickness can be changed using:

```
((Graphics2D)g).setStroke(new BasicStroke(5));
```

**Create a class called Robot class that has the following attributes:**

### Variables

- Box leftLeg, rightLeg, body, leftArm, rightArm, head, rightEye, leftEye, mouth – the boxes that make up the face of the Robot
- boolean armsUp

### Methods

Name: Robot

Input: Box leftLeg, Box rightLeg, Box body, Box leftArm, Box rightArm, Box head, Box rightEye, Box leftEye, Box mouth

Output: nothing

Action: a constructor that takes in the initial values of the class variables. armsUp should be set to false.

Name: getBox

Input: String whichPart

Output: Box head

Action: returns the Box that corresponds to the body part passed in. The options include “Left Leg”, “Right Leg”, “Body”, “Left Arm”, “Right Arm”, “Head”, “Right Eye”, “Left Eye”, “Mouth”

Name: moveRight

Input: int amount

Output: nothing

Action: moves the robot to the right by the specified amount. This should be done by moving each of the boxes to the right.

Name: moveLeft

Input: int amount

Output: nothing

Action: moves the robot to the left by the specified amount. This should be done by moving each of the boxes to the left.

Name: moveUp

Input: int amount

Output: nothing

Action: moves the robot up by the specified amount. This should be done by moving each of the boxes up.

Name: moveDown

Input: int amount

Output: nothing

Action: moves the robot down by the specified amount. This should be done by moving each of the boxes down.

Name: closeLeftEye

Input: nothing

Output: nothing

Action: changes the height of the left eye to zero

Name: openLeftEye  
Input: int amount  
Output: nothing  
Action: increases the height of the left eye to amount times 2

Name: closeRightEye  
Input: nothing  
Output: nothing  
Action: changes the height of the right eye to zero

Name: openRightEye  
Input: int amount  
Output: nothing  
Action: increases the height of the right eye to amount times 2

Name: growLegs  
Input: int amount  
Output: nothing  
Action: increases the height of the legs upward by amount. All other parts should be moved upward by amount.

Name: swivelArms  
Input: nothing  
Output: nothing  
Action: switches the arms from being up to down or down to up depending on what they were before. Also updates the armsUp boolean. The arms should be swiveled up by stretching up by height and stretching down by half the width. Do the opposite to swivel arms down.

Name: paint  
Input: Graphics  
Output: nothing  
Action: draws the Robot to the screen by telling each box to paint itself. The head should be painted before the eyes and mouth.