Chapter 8

Classes and Objects
Data Abstraction

• In Java there are three types of data values:
  – primitive data values (int, float, boolean, etc.)
  – arrays (actually a special type of object)
  – objects

• Objects in a program are used to represent "real" (and sometimes not so real) objects from the world around us.
Objects

- An object might represent a string of characters, a planet, a type of food, a student, an employee, a piece of email, a ball, a car, a font, an image,... anything that can't be (easily) represented by a primitive value or an array.
- Just as 3 is a primitive value of type int, every object must also have a type. These types are called classes.
Classes

A class describes a set of objects.

• It specifies what information will be used to represent an object from the set (e.g. name and salary for an employee, position, velocity, and size for a ball).

• It also specifies what operations can be performed on such an object (get the name of the student, send an email message, move the ball).
Working with Objects

• Use new to make a new object e.g.
  Ball ball1 = new Ball(x,y,xv,yv,dia);
  Car car1 = new Car(top,left);
  String name = “Dustin”;

• Operate on them with “dot notation”.
  ball1.update();
  car1.move();
  String fullName = name.append(“Adams”);
Which does not belong in this list considering what was on the last slide?

A. int
B. String
C. Car
D. name
E. Ball

• Use new to make a new object e.g.
  Ball ball1 = new Ball(x,y,xv,yv,dia);
  Car car1 = new Car(top,left);
  String name = “Dustin”;

• Operate on them with “dot notation”.
  ball1.update();
  car1.move();
  String fullName = name.append(“Adams”);
// our old ball example from before
float velocity = 0;
float yPos = 0;
int ballRadius = 10;
float gravity = 0.1;
void setup() {
    size(400,400);
    fill(0); // ball will be black
}
void draw() {
    background(255);
    // update the ball position and velocity
    if (yPos > height-ballRadius) {
        velocity = -velocity;
    }
    yPos = yPos + velocity;
    velocity = velocity + gravity;
    velocity = velocity*0.99; // add some drag
    // draw the ball
    ellipse(width/2, yPos, ballRadius*2, ballRadius*2);
}

// What would it take to add a second ball?
Ball ball1;
float gravity = 0.1;
float drag = 0.99;
void setup() {
    size(400, 400);
    ball1 = new Ball(width/4, 0, 0, 0, 0, 10);
    fill(255, 0, 0);
}
void draw() {
    background(255);
    ball1.update();
}
Ball ball1, ball2;
float gravity = 0.1;
float drag = 0.99;
void setup() {
    size(400, 400);
    ball1 = new Ball(width/4, 0, 0, 0, 0, 10);
    ball2 = new Ball(3*width/4, 0, 0.5, 0.3, 20);
    fill(255,0,0);
}
void draw() {
    background(255);
    ball1.update();
    ball2.update();
}
class Ball {
    float x, y, xVel, yVel;
    int diam;

    void update() {
        ...
    }

    ...
}
void update() {
    // if hit the ground so need to reverse the velocity
    if (y > height-diam/2) {
        yVel = -yVel;
    }
    // adjust position based on velocity
    y = y + yVel;
    x = x + xVel;

    // adjust the velocity - increasing due to gravity
    yVel = yVel + gravity; // always accelerate down

    // add some drag
    yVel = yVel*drag;

    // draw the ball
    ellipse(x, y, diam, diam);
}
class Ball {
    float x, y, xVel, yVel;
    int diam;
    void update() {
        ...
    }
    Ball(float xStart, float yStart, float xV, float yV, 
     int diameter) {
        x = xStart;
        y = yStart;
        xVel = xV;
        yVel = yV;
        diam = diameter;
    }
}
Given the Ball code just presented, which of these statements is not legal?

A. ball = new Ball(10, 20, 0, 0, 0);
B. ball = new Ball(0, 0, 0, 0, 0);
C. ball = new Ball();
D. ball.update();
E. ball.diam = 200;

class Ball {
    float x, y, xVel, yVel;
    int diam;
    void update() {
        ...
    }
    Ball(float xStart, float yStart, float xV, float yV,
         int diameter)
    {
        x = xStart;
        y = yStart;
        xVel = xV;
        yVel = yV;
        diam = diameter;
    }
}
Elements of a Simple Class

• A class describes the data values used to represent an object and any operations that can be performed on that object.

• The data values are stored in *instance variables*, also known as *fields*, or *data members*.

• The operations are described by *instance methods*, sometimes called *procedure members*.
How many operations were defined for the Ball class?

Do not count creating the Ball as an operation on a Ball.

A. 0

B. 1

C. 2

class Ball {
    float x, y, xVel, yVel;
    int diam;
    void update() {
        . . .
    }
    Ball(float xStart, float yStart, float xV, float yV,
         int diameter)
    {
        x = xStart;
        y = yStart;
        xVel = xV;
        yVel = yV;
        diam = diameter;
    }
}
Making Your Own Classes

class Ball {
    float x, y, xVel, yVel;
    int diam;
}

// redone with a simple ball class - no methods
Ball ball1;
float gravity = 0.1;
void setup() {
  size(400, 400);
  ball1 = new Ball();
  ball1.x = width/4;
  ball1.y = 0;
  ball1.diam = 10;
  ball1.xVel = 0;
  ball1.yVel = 0;
  fill(255,0,0);
}
void draw() {
    background(255);
    // update the ball position and velocity
    if (ball1.y > height-ball1.diam/2) {
        ball1.yVel = -ball1.yVel;
    }
    ball1.y = ball1.y + ball1.yVel;
    ball1.yVel = ball1.yVel + gravity;
    ball1.yVel = ball1.yVel*0.99; // add some drag

    // draw the ball
    ellipse(ball1.x, ball1.y, ball1.diam, ball1.diam);
}
What is needed to add another ball?
Modify the Ball class to allow for specifying the ball color.

Add a method so that the Ball color can be changed repeatedly after creation/initialization.

Fix update to update both x and y positions and velocities.

Add a method that takes a Ball as a parameter and returns true if the “other” ball has bumped into “this” ball. What other operations might you add to the Ball class?
Elements of a Simple Class

• A class describes the data values used to represent an object and any operations that can be performed on that object.

• The data values are stored in **instance variables**, also known as **fields**, or **data members**.

• The operations are described by **instance methods**, sometimes called **procedure members**.
class Ball {
    float x, y, xVel, yVel;
    int diam;
    void update() {
        . . .
    }
    Ball(float xStart, float yStart) {
        x = xStart;
        y = yStart;
        xVel = 0;
        yVel = 0;
        diam = 20;
    }
}
class Ball {
    float x, y, xVel, yVel;
    int diam;
    void update() {
        ...  
    }
    Ball(float xStart, float yStart, float xV, float yV,
         int diameter)
    {
        ...
    }
    boolean collidedWith(Ball other) {
        ...
    }
    void reverseX() {
        x = -x;
    }
}
class Ball {
    float x, y, xVel, yVel;
    int diam;
    color bColor;
    void update() {
        // draw the ball
        fill(bColor);
        ellipse(x, y, diam, diam);
    }
}

Ball(float xStart, float yStart, float xV, float yV, int diameter, color bColor) {
    x = xStart;
    y = yStart;
    xVel = xV;
    yVel = yV;
    diam = diameter;
}

Are the boldface additions sufficient to assign different colors to balls?
A. Yes
B. No - there is a missing line.
C. No - there is a syntax error.
D. No - there are 2 missing lines.
// We can have more than one Constructor

class Ball {
    float x, y, xVel, yVel;
    int diam;
    void update() {
        . . .
    }
    Ball() {
        . . .
    }
    Ball(float xStart, float yStart) {
        . . .
    }
    Ball(float xStart, float yStart, float xV, float yV, int diameter) {
        . . .
    }
}
Create a class to represent a light bulb that can be turned on and off.

```
Light light1, light2, light3;
void setup() {
    size(400, 400);
    light1 = new Light(25, 25);
    light2 = new Light(width-30, height-60, false);
    light3 = new Light(width/2, height/2, 50);//specify size
}

void draw() {
    light1.show();
    light2.show();
    light3.show();
}
```

What’s the minimum number of instance variables the Light class can have?

A. 2  B. 3  C. 4  D. 5  E. 6
void mousePressed() {
    if (light1.clicked()) {
        light1.toggle();
    }
    if (light2.clicked()) {
        light2.toggle();
    }
    if (light3.clicked()) {
        light3.toggle();
    }
}

// How many methods plus constructors must the Light class have?
A. 2  B. 3  C. 4  D. 5  E. 6

//Create a class to represent a light bulb that can be turned on and off.
Light light1, light2, light3;
void setup() {
    size(400, 400);
    light1 = new Light(25, 25);
    light2 = new Light(width-30, height-60, false);
    light3 = new Light(width/2, height/2, 50);//specify size
}
void draw() {
    light1.show();
    light2.show();
    light3.show();
}