Introduction to OOP

COMP2026

PROBLEM SOLVING USING OBJECT ORIENTED PROGRAMMING

Overview

- Why objects?
- Developing our own object types

Why objects?

Bees

- Suppose you are told to write a mini-game with 10 bees flying in the sky at different speeds…

Bees

- You may think of …

```
...
//x and y coordinates of the bees
int x1, x2, x3, x4, x5, x6, x7, x8, x9, x10;
int y1, y2, y3, y4, y5, y6, y7, y8, y9, y10;
...
//horizontal velocities of the bees
int dx1, dx2, dx3, dx4, dx5,
    dx6, dx7, dx8, dx9, dx10;
...

//move the bees horizontally
x1 = x1 + dx1;
x2 = x2 + dx2;
x3 = x3 + dx3;
...
```

Bees

- A better way …

```
...
//x and y coordinates of the bees
int[] x = new int[10];
int[] y = new int[10];
...
//horizontal velocities of the bees
int[] dx = new int[10];
...

//move the bees horizontally
for (int i = 0; i < x.length; i++)
{
   x[i] = x[i] + dx[i];
}
...
```

Bees

- How about having bee objects?

```
x = 55;
y = 25;
dx = 5;
void moveHorizontally()
{
 x = x + dx;
}
```

```
x = 78;
y = 97;
dx = 3;
void moveHorizontally()
{
 x = x + dx;
}
```

```
x = 8;
y = 65;
dx = 10;
void moveHorizontally()
{
 x = x + dx;
}
```

Each bee object has its own values (x, y, and dx)

The moveHorizontally() method updates the values in its own bee object

Bees

- With bee objects…

```
//Create bees with x, y, and speed
Bee b1 = new Bee(55, 25, 5);
Bee b2 = new Bee(78, 97, 3);
Bee b3 = new Bee(8, 65, 10);
...

//move the bees horizontally
b1.moveHorizontally();
b2.moveHorizontally();
b3.moveHorizontally();
...
```

Bees

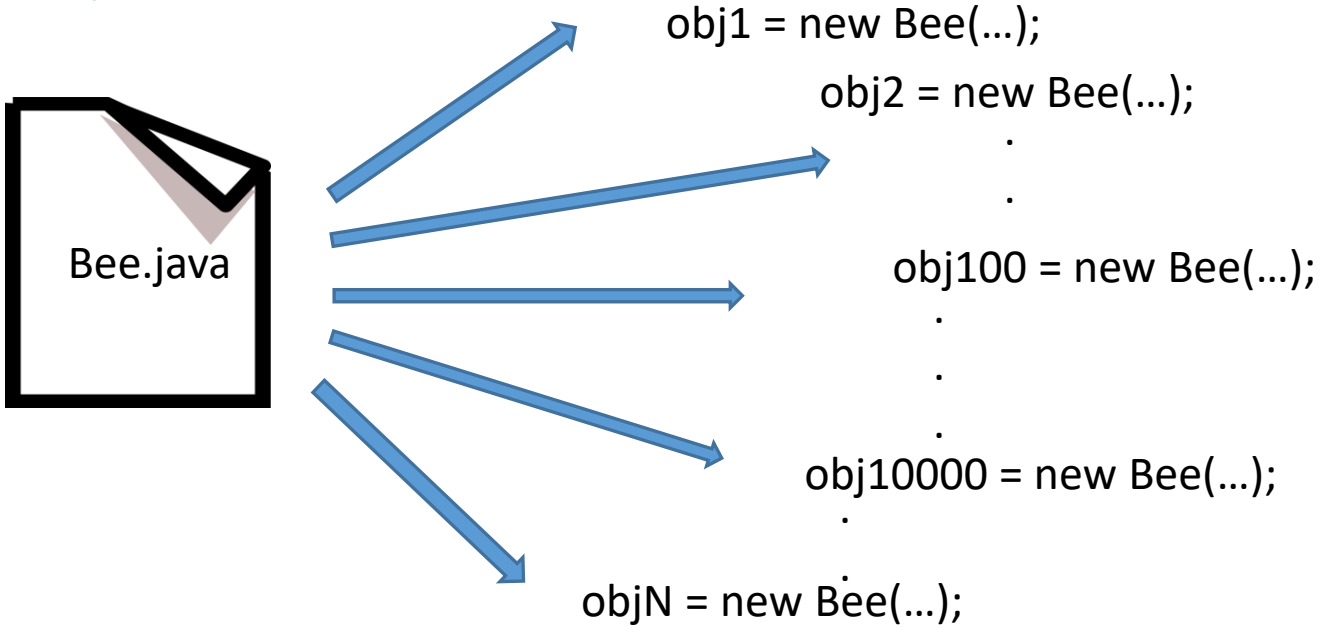
- Array of bee objects…

```
//array to store 10 bee objects
Bee[] bees = new Bee[10];
for (int i = 0; i < bees.length; i++)
{
   bees[i] = new Bee(...);
}
...

//move the bees horizontally
for (int i = 0; i < bees.length; i++)
{
   bees[i].moveHorizontally();
}
...
```

A class is a blueprint of the object

- A class is a blueprint that defines the variables and the methods common to all objects of a certain kind
- Only write one Bee class, and we can create lots of Bee objects



class Bee

```
public class Bee {
    private int x; //x-coordinate
    private int y; //y-coordinate
    private int dx; //horizontal velocity

    //Constructor to create bee object
    public Bee(int x, int y, int dx) {
        this.x = x;
        this.y = y;
        this.dx = dx;
    }

    //move this bee horizontally
    public void moveHorizontally() {
        x = x + dx;
    }
    ...
```

Fields/Instance variables to store the state of the object. Each object has its own set of fields/instance variables.

Note: When we use the **new** operator to create object, we are calling the constructor.

```
Bee obj = new Bee(10, 20, 5);
```

constructor

Instance methods to define the behavior of the object.

Example of usage:

```
obj.moveHorizontally();
```

class Bee

```
public int getX() {
    return x;
}
public void setX(int x) {
    this.x = x;
}
public int getY() {
    return y;
}
public void setY(int y) {
    this.y = y;
}
public int getDx() {
    return dx;
}
public void setDx(int dx) {
    this.dx = dx;
}
```

Get and set methods to retrieve and update the fields/instance variables after the object is created.

Examples of usage:

```
int x = obj.getX();
obj.setX(100);
```

Example on creating objects

```
public static void main(String[] args) {
  //create Bee objects
  Bee alice = new Bee(55, 25, 5);
  Bee bob = new Bee(78, 97, 3);

  //print the objects
  System.out.println("Alice: " + alice);
  System.out.println("Bob: " + bob);
  System.out.println();

  alice.moveHorizontally(); //move the bees
  bob.moveHorizontally ();  //horizontally

  //print the objects
  System.out.println("After moving horizontally:");
  System.out.println("Alice: " + alice);
  System.out.println("Bob: " + bob);
  System.out.println();
  ...
```

Example on creating objects

```
...
//update the x, y and dx of an object
alice.setX(100);
alice.setY(200);
alice.setDx(30);

//get the x, y and dx from an object
int aliceX = alice.getX();
int aliceY = alice.getY();
int aliceDx = alice.getDx();

System.out.println("After update:");
System.out.println("Alice's X: " + aliceX);
System.out.println("Alice's Y: " + aliceY);
System.out.println("Alice's horizontal velocity: " +
                                          aliceDx);
System.out.println();
...
```

Part A
Discovery Exercises

Type your answer in **XXXXXXXX_lab08.docx**

Part B
Programming Exercises

Hints for Task 1

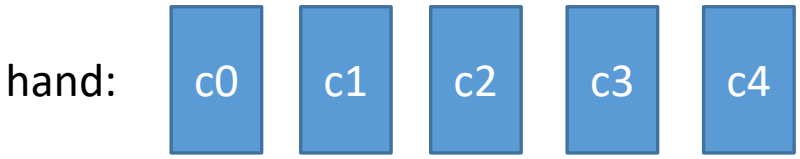
- How to convert the integer suit and rank to Strings?
  - Use String array

```
String[] s = {"Diamonds" , "Clubs", "Hearts", "Spades"};
```

  - **`s[suit]`** will give out the String
  - Create a similar String array for rank

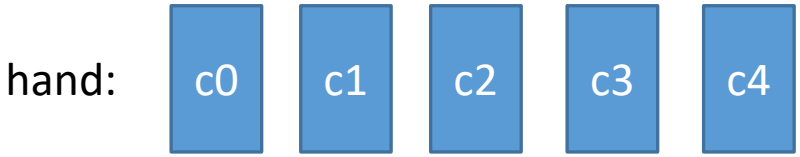
Hints for Task 2

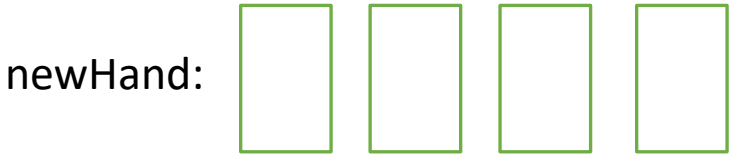
- How to remove a card from the hand?
  - Suppose we want to remove the card c2



Hints for Task 2

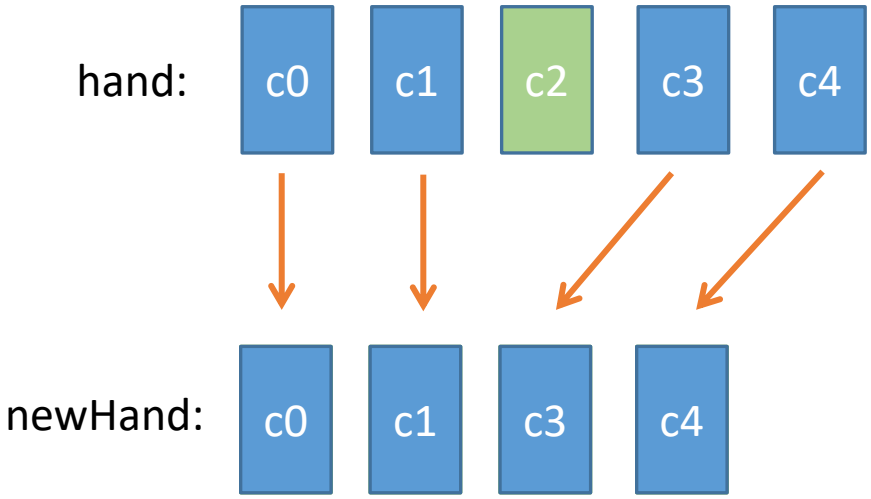
- Create a new array of cards, say newHand, with size **`hand.length-1`**





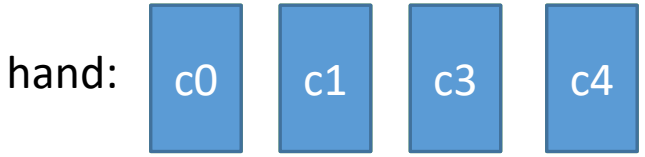
Hints for Task 2

- Copy all the cards, except c2, from hand to newHand



- Then, assign newHand to hand

```
hand = newHand
```



Lab Exercise Submission

- Submit the following to Moodle
  - ***XXXXXXXX*_ lab08.docx**
  - ***XXXXXXXX*_lab08.zip**

*Replace "***XXXXXXXX***" with your **student ID**

Deadline: Before next Monday noon

References

- Dean, J., & Dean, R. (2008). Introduction to programming with Java: A problem solving approach. Boston: McGraw-Hill.
- Forouzan, B. A., & Gilberg, R. F. (2007). Computer science: A structured programming approach using C (3rd ed.). Boston, MA: Thomson Course Technology.
- Gaddis, T. (2016). Starting out with Java (6th ed.). Pearson.
- Liang, Y. D. (2013). *Introduction to Java programming: Comprehensive version*. (8th ed.). Pearson.
- Schildt, H. (2006). *Java a beginner's guide*. New York: McGraw Hill.
- Wu, C. T. (2010). *An introduction to object-oriented programming with Java*. Boston: McGraw Hill Higher Education
- Xavier, C. (2011). Java programming: A practical approach. New Delhi: Tata McGraw Hill.
- Zakhour, S., Kannan, S., & Gallardo, R. (2013). The Java tutorial: A short course on the basics (5th ed.).
- yet another insignificant Programming Notes. (n.d.). Retrieved from https://www3.ntu.edu.sg/home/ehchua/programming