

Topics: Sorting, array of objects, **this** revisited, concise code

Reading (JV): Sec 6.2, "Selection Sort" in Sec 6.3

Template for selection sort (ascending order)

```
// loop from first to second last element

    // find index of minimum value _____

// swap ith element with minimum value
```

Array of objects

- Elements of an array can be object references
- Three steps: (1) declaration of the array reference variable, (2) creation (instantiation) of the array of object references, and (3) instantiation of individual objects
- E.g., the following statement reserves space to store 10 **Room** references (assuming a **Room** class is defined):

```
Room[] caves = new Room[10];
```

The individual Room objects need to be created separately:

```
caves[0] = new Room();
caves[1] = new Room();
...
```

Example

```
/* Organize data for any Person:  name, age */
public class Person {
    private String name;
    private int age;

    // Constructor
    public Person(String n, int a) {
        name = n;
        age = a;
    }

    // Determine if self is an adult
    public boolean isAdult() {
        final int mature = 18;
        return age >= mature;
    }

    // Show object data
    public String toString() {
        return name + " is " + age;
    }
} // class Person
```

Alternate constructor

*Inconcise code for boolean type that you should **not** write!*

```

/* Client class that uses Person class: create a collection of Person data */
public class Record {
    public static void main(String[] args) {
        int size = 100;        // max length of record

        // declare reference variable for array (of Person objects)

        // instantiate array of Person references

        // create Person objects
        record[0] = new Person("Daisy", 19);
        record[1] = new Person("Rob", 18);
        record[2] = new Person("Mary", 16);

        // report only the adults
        for (int i=0; i<3; i++)
            if ( record[i].isAdult() )
                System.out.println(record[i]);
    } // method main
} // class Record

/* Client class of Person class that reads user input to create Person objects */
import cs1.Keyboard;
public class Record2 {
    public static void main(String[] args) {
        int size = 100;        // max length of record
        int count = 0;         // # entries so far
        String name;           // a person's name
        int age;               // a person's age
        Person[] record;       // ref variable for array of Person objects

        // instantiate array of Person references, length $size$
        record = new Person[size];

        // read data and create Person objects
        String prompt = "Enter on separate lines name and age";
        prompt += ", type * <Enter> -1 <Enter> to end";
        System.out.println(prompt);
        name = Keyboard.readString();
        age = Keyboard.readInt();
        while (age>=0) {
            record[count] = new Person(name, age);
            count++;
            name = Keyboard.readString();
            age = Keyboard.readInt();
        }

        // report only the adults
        for (int i=0; i<count; i++)
            if ( record[i].isAdult() )
                System.out.println(record[i]);
    } // method main
} // class Record2

```

Beware of null references

```

// Suppose we loop through entire
// array. Then we must first check
// for existence of object BEFORE
// accessing an object's instance
// method
for (int i=0; i<size; i++)
    if ( record[i]!=null &&
        record[i].isAdult() )
        System.out.println(record[i]);

```