Overloading Methods

1. public Car(String licensePlate1, double maxSpeed1, double speed1) {
   this.licensePlate = licensePlate1;
   this.speed = 0.0;
   if (maxSpeed1 >= 0.0) {
      maxSpeed = maxSpeed1;
   } else {
      maxSpeed = 0.0;
   }
   // set speed according to the value of passed argument
   if (speed1 < 0)
      speed = 0;
   else if (speed1 > maxSpeed)
      speed = maxSpeed;
   else
      speed = speed1;
}

2. // print the details of the car
   public void print() {
      System.out.println("Current speed: " + speed + "\nMax speed: " + maxSpeed +
   }

3. public class CarTest {
   public static void main(String[] args) {
      Car car1 = new Car("K123WMI", 150.0);
      car1.print();

      Car car2 = new Car("L777ALA", 120.0, 80.0);
      car2.print();

      Car car3 = new Car("F180AST", 180.0, 40.0);
      car3.print();
   }
}
Garbage Collection

At the point where the garbage collector runs, a1 is null, a2, a3 and a4 point to object 1, a5 is null.

Therefore objects 2 and 3 will be recycled by the garbage collector, as there are no more references to them. Object 1 will remain in the heap unaffected.

Using arrays to collect statistics

```java
import java.util.*;

public class StatsInspector {
    /* the following array stores in each element how many times
     the given element is generated by the random generator.
     Thus, outcome[0] corresponds to how many times value 0 has
     been generated, outcome[1] how many times 1 has been
     generated, etc. */
    private int outcome[];

    /* print a histogram showing how many times each value
     was generated */
    public void printHistogram() {
        // print how many times each value was generated
        for (int i=0; i < outcome.length; i++) {
            System.out.print(i + " : ");
            // print a number of stars corresponding to value
            for (int j=1; j <= outcome[i]; j++)
                System.out.print("*");
            // insert a newline after each row is printed
            System.out.println();
        }
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("How many random numbers should be generated? ");
        int numbers = sc.nextInt();
        System.out.print("What is the number of values for each random draw? ");
        int values = sc.nextInt();

        StatsInspector stats = new StatsInspector();
        stats.outcome = new int[values];

        // create an instance of a random generator
        Random randGenerator = new Random();
```
// call repeatedly "numbers" times the random generator
for (int i=1; i <= numbers; i++) {
    int n = randGenerator.nextInt(values);
    System.out.println("Generated: "+n);
    stats.outcome[n]++;  // increase corresponding counter by one
}

// print how many times each value was generated
for (int i=0; i < values; i++)
    System.out.println(i + " "+ stats.outcome[i]);

// print the histogram for the generated values
stats.printHistogram();
}

Arrays of Objects

1. import java.util.Scanner;

    class Student {
        String name;
        double average;
    }

    public class StudentStats {
        Student students[] = new Student[50];  // assume max of 50 students
        int number_of_students;

        void readAverages() {
            String name = "";
            Scanner sc = new Scanner(System.in);

            // keep reading students until the user inputs 'q'
            while (!name.equals("q")) {
                System.out.print("Student name: ");
                name = sc.next();

                // Exit loop is user entered 'q'
                if (name.equals("q"))
                    break;

                System.out.print("Average grade: ");
                double average = sc.nextDouble();
                Student st = new Student();
                st.name = name;
                st.average = average;
            }
        }
    }
2. The two new methods are inserted in the above code. The complete code for the class is:

```java
import java.util.Scanner;

class Student {
    String name;
    double average;
}

public class StudentStats {
    Student students[] = new Student[50]; // assume max of 50 students
    int number_of_students;

    void readAverages() {
        String name = "";
        Scanner sc = new Scanner(System.in);

        // keep reading students until the user inputs 'q'
        while (!name.equals("q")) {
            System.out.print("Student name: ");
            name = sc.next();

            // Exit loop is user entered 'q'
            if (name.equals("q"))
                break;

            System.out.print("Average grade: ");
            double average = sc.nextDouble();
            Student st = new Student();
            st.name = name;
            st.average = average;

            // store Student object in array students
            if (number_of_students < students.length)
                students[number_of_students++] = st;
            else {
                System.out.println("Cannot store any more students");
                break;
            }
        }
    }
```
String findBest() {
    double max = 0.0;
    int max_index = -1; // index of student with largest average
    for (int i=0; i < number_of_students; i++)
        if (max < students[i].average) {
            max_index = i;
            max = students[i].average;
        }
    if (max_index >= 0)
        return students[max_index].name;
    else
        return "No students found!";
}

// find the student with worst average and returns his/her name
String findWorst() {
    double min = Double.MAX_VALUE;
    int min_index = -1; // index of student with largest average
    for (int i=0; i < number_of_students; i++)
        if (min > students[i].average) {
            min_index = i;
            min = students[i].average;
        }
    if (min_index >= 0)
        return students[min_index].name;
    else
        return "No students found!";
}

public class StudentStatsTest {
    public static void main(String[] args) {
        StudentStats stats = new StudentStats();
        stats.readAverages();

        // find student with best average
        String best_student = stats.findBest();
        System.out.println("Student with max average: " + best_student);

        // find student with worst average
        String worst_student = stats.findWorst();
        System.out.println("Student with min average: " + worst_student);
    }
}
The Arrays class

import java.util.*;

public class ArraysTest {
    int numbers[] = new int[50]; // assume 50 integers max
    int index; // holds how many numbers are stored in array numbers

    /* sort integers in numbers array and displays them in ascending order */
    void display() {
        // only sort elements between 0 and index exclusive
        Arrays.sort(numbers, 0, index);

        // printed sorted elements
        for (int i=0; i < index; i++)
            System.out.println(numbers[i]);
    }

    public static void main(String[] args) {
        ArraysTest a = new ArraysTest();

        int number = 0;
        Scanner sc = new Scanner(System.in);

        // keep asking user for numbers until entering 9999
        while (number != 9999) {
            System.out.print("Enter number (9999 to quit): ");
            int n = sc.nextInt();
            if (n != 9999)
                a.numbers[a.index++] = n;
            else
                break;

            if (a.index >= a.numbers.length)
                break;
        }

        // display numbers in ascending order
        a.display();
    }
}