1. (a) int temp = b; b = c; c = temp;

1. (b) The four kinds of variable are:
(1) parameter (declared within the parentheses of a method header)
(2) local variable (declared within a method body)
(3) instance variable or field (declared within the a class body)
(4) static or class variable (declared with attribute static within a class body).

1 (c). It is not possible to write such a procedure. When a frame for the given call is created, the values of b and c (or c and b) are stored in parameters x and y, and b and c cannot be referenced in the method body. Drawing the frame for the call, as shown in part 1(d), should give you insight into this issue.

1(d).

<table>
<thead>
<tr>
<th>Swap: 1</th>
<th>Cornellian</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>y</td>
</tr>
</tbody>
</table>

2. (a) /** Instance maintains info about a Cornellian */
public class Cornellian {
private String name; // this student's name
private static int nextID = 0;
/** Constructor: a Cornellian with name name */
public Cornellian(String name) {
    this.name = name;
}
/** = String repr. of Cornellian, giving just the name */
public String toString() {
    return name;
}
/** An instance maintains info about a student */
public class Student extends Cornellian {
private double gpa; // student's GPA
/** Constructor: a student with name n and GPA g */
public Student(String n, double g) {
    super(n);
    gpa = g;
}
/** = this student's GPA */
public double getGPA() {
    return gpa;
}
/** = "this student has GPA >= 3.5 " */
public boolean isOnDeansList() {
    return gpa >= 3.5;
}
/** = a representation of this student. Form:
student's name, followed by " Dean's List" if this student is on the Deans List. */
public String toString() {
    return super.toString() +
    (isOnDeansList() ? " Dean's List": "") +
    Cornellian.incrementId();
}
}

3. (a) /** = a unique positive integer. (Every call on incrementId gives a different integer). */
public static int incrementId() {
    nextID = nextID + 1;
    return nextID;
}

4. (b) int v = Cornellian.incrementId();

4. /** = a netid for name s. Precondition: s has the form given in the box above and to the right on page 3 of Prelim I. */
public static String netId(String s) {
    s = s.trim().toLowerCase();
    int k = s.indexOf(" ");
    String lastName = s.substring(0, k).trim();
    // Assume middle name is missing
    String firstName = s.substring(k + 1).trim();
    String middleName = "";
    k = firstName.indexOf(" ");
    if (k > 0) {
        middleName = firstName.substring(k).trim();
        firstName = firstName.substring(0, k);
    }
    return "" + firstName.charAt(0) +
    (middleName.length() > 0 ?
    middleName.charAt(0) : "") +
    lastName.charAt(0) +
    incrementId();
}