ANSWERS

1.
```
/** Sort array b, using selection sort */
public void selectionSort(int[] b) {
  int k= 0;
  // inv: b[0..k-1] is sorted and
  //      b[0..k-1] ≤ b[k..]
  while (k != b.length) {
       Swap b[k] with smallest of
              b[k..b.length-1];
       k= k+1;
    }
    // b[0..b.length-1] is sorted
}
```

2 .

(a) Inmate.compare(x,y);

(b) v.add(**new** Inmate(42)); or
v.addElement(**new** Inmate(42));

(c) Object fred= v.get(0);
The apparent type is Object.

(d) **int** x= ((Inmate) fred).getID();

3.
```
// inv: lineup[0..i-1] is in ascending order
  for (int i= 1; i < lineup.length; i= i+1) {
      if (lineup[i-1].getID() > lineup[i].getID())
          return false;
  }
  // lineup[0..lineup.length-1] is in ascending order
  return true;
```

4.
```
/** An instance is a red-nosed clown */
public class RedNosedClown extends Clown {
    /** Constructor: red nosed clown with id id */
    public RedNosedClown(int id) {
        super(id);
    }

    /** = the kind of this Clown */
    public String theClownyThing() {
        return "Well, I have a red nose";
    }

    /** = new Clown with same kind and id
            as this one */
    public Clown duplicateSelf() {
        return new RedNosedClown(getID());
    }
}
```

(5a)
```
/** Remove all Clowns from train[i..] that have
    the same id as Clown target.
    Precondition: 0 ≤ i ≤ train.size() */
public static void ejectClown
                (Vector train, Clown target, int i) {
    int k= i;
    // inv: no clown in train[i..k-1] has the same id
    //      as target
    while (k != train.size()) {
        Clown suspect= (Clown)(train.get(k));
        if (Inmate.compare(suspect,target))
            train.remove(k);
        else
            k= k+1;
    }
    // no clown in train[i..train.size()-1] has the same id
    //  as target
}
```

(5b)
```
/** Remove duplicates from train */
public static void purgeTrain(Vector train) {
    int h= 0;
    // inv: Duplicates of train[0..h-1] have been
    //      removed from train
    while (h != train.size()) {
        // Remove from train[h+1..] Clowns
        //   with same id as train[h]
        ejectClown(train,(Clown)(train.get(h)), h+1);

        h= h+1;
    }
    // Duplicates of train[0..train.size()-1] have been
    // removed from train
}
```