

Department of Computer Science, UIUC
 CS125 Second Examination - SOLUTION

```
1. (a) Robot r1, r2, r3;
    r1 = new Robot("Huey", 2,3);
    r2 = new Robot("Louie", 5,4); <- r1
    r3 = new Robot("Louie", 5,4); <- r2
    r1 = r2;
    r2 = r3;
    r3 = r1;
```

System.out.print(r1 == r2); Write exactly the program output here and nothing else: **false**

(b) Write A or B here: **B**

(c) Your answer: **D (Polymorphism)**

(d) Your answer: **true**

(e) **The class includes methods which change instance variables e.g. a setter. (Providing a constructor that sets the instance variable is not a good example)**

2. Be careful when comparing strings – 16 points

```
public class FriendsList {
    private String[] list = new String[0];
    public void add(String name) {
        //Allocate a larger array:
        String[] tmp = new String[list.length + 1];
        //Copy the old list contents :
        for(int i=0;i<list.length) tmp[i] = list[i];
        //update this.list to reference the new array:
        list=tmp;
        //Don't forget to add the new friend:
        list[list.length-1] = name; //list == tmp at this point
    }
    public int getSize() { return list.length;}
    public String getFriend(int i) {return list[i];}
    public boolean is007() {
        for(int i=0;i<list.length;i++)
            if( list[i].equals("James Bond") ) return true;
        return false;
    }
} // end class FriendsList
```

3.

srcA	x=0	x=1	x=2
y=0			
y=1			
y=2			
y=3			

srcB	x=0	x=1	x=2
y=0			
y=1			
y=2			
y=3			

output	x=0	x=1	x=2
y=0			
y=1	X	X	X
y=2			X
y=3		X	X

(b) When tested with two test images for *srcA* *srcB*, the function fails with an *IndexOutOfBoundsException* at line 6.

srcB is not as wide as srcA srcB.length < srcA.length srcB has fewer columns

(c) The function fails with an exception at line 2. Suggest a possible parameter value

srcA is null

(d)

```
for(int y=0;y<height;y++)
    output[x][y] = srcA[x][y] ==0 ? srcB[x][y] : srcA[x][y];
```

```
4. public class Boat {
    private int x,y,count;
    public Boat(int xx, int yy) { x = xx;y=yy; count=3;}
    public Boat(Boat other) { x=other.x;y=other.y;count = other.count;}
    public boolean turn(Point p) {
        if(p.x == x && p.y==y) {count--; return true;}
        return false;}
    public boolean isAlive() {return count >=0;}
}
```

```
public class Point {
    public int x;
    public int y;
}
```

```
5. public class Submarine extends Boat {
    private boolean submerged;
    public Submarine() {
        super(5,6);
        submerged = Math.random() >= 0.5;
    }
    public boolean turn(Point p) {
        if(submerged) return false;
        return super.turn(p);
    }
    public void toggle() {
        submerged = ! submerged;
    }
}

6. public class BoatUtil {
    public static Boat[] create(int n) {
        Boat[] result= new Boat[n];
        for(int i=0;i<n;i++)
            result[i] = new Boat(i,i);
        return result;
    }
    public static Boat[] boatcopy(Boat[] source) {
        Boat[] result = new Boat[ source.length ];
        for(int i=0;i< source.length;i++)
            result[i] = new Boat(source[i]);
        return result;
    }
}

public class BattleshipGame {
    public static void main(String[] args) {
        Boat[] b = BoatUtil.create(7);
        Boat[] b2 = BoatUtil.boatcopy(b);
        b2[0].turn( new Point() );
    }
} // end of class (END OF EXAM)
```