CPSC 233 Midterm Review Questions, Fall 2003

Multiple choice questions

For each question make sure that you select the **best** answer to that question

1) Suppose that the bit pattern of an integer "num" starts out as 000…0001 (all zeros except for the last bit which is a one).   What will the bit pattern look like after the following statement: num = ~num;
   a) 000…0000
   b) 000…0001
   c) 000…0010
   d) 111....1110
   e) None of the above

d

2) Overloaded methods have the same name. Which of the following criteria are used to distinguish overloaded methods?
   a) The number of parameters for each method.
   b) The return type of the method.
   c) The type of each parameter
   d) (a) & (c)
   e) All of the above.

d

3) Which Object-Oriented concept(s) is exemplified by the following class definition?

```
class Foo
{
    private int num;
    public int getNum () { return num; }
    public void setNum (int newValue)
    {
        if ((newValue < 0) || (newValue > 100))
            System.out.println("Num can only be set to values from 0
                - 100");
        else
            num = newValue;
    }
}
```

   a) Information hiding
   b) Implementation hiding
   c) Method overloading
   d) More than one of the above are true
   e) None of the above

a

4) Assuming that the default initialization value for integers is zero, what will be the output of the following program?

```
class Q4
{
    public static void main (String [] args)
    {
        final int SIZE = 5;
        int i;
        int [] arr = new int [SIZE];
        for (i = 0; i < SIZE; i++)
            arr[i] = i;
        arr = new int[SIZE];
        for (i = 0; i < SIZE; i++)
            System.out.print(arr[i]);
    }
}
```

a.   0
b.   4
c. 00000
d. 01234
e. None of the above answers are true.

c

Short Answer questions

1. For this question please refer to the code and code fragments from the Biosphere & Critter classes from the Game of Life:

```
class Biosphere
{
    private Critter [][] previous;
    private Critter [][] current;
        :     :     :     :
    public void copy ()
    {
        int r, c;
        for (r = 0; r < ROWS; r++)
        {
            for (c = 0; c < COLUMNS; c++)
            {
                previous[r][c] = current[r][c];
            }
        }
    }
}

class Critter
{
    private char appearance;
    public Critter ()
    {
        appearance = '*';
    }
```

```
    public Critter (char ch)
    {
        appearance = ch;
    }

    public char getAppearance ()
    {
        return appearance;
    }

    public void setAppearance (char newAppearance)
    {
        if ((newAppearance == ' ') || (newAppearance == '*'))
            appearance = newAppearance;
        else
            System.out.println("Critter's appearance must be either a
                star or a space.");
    }

    public void display ()
    {
        System.out.print(appearance);
    }
}
```
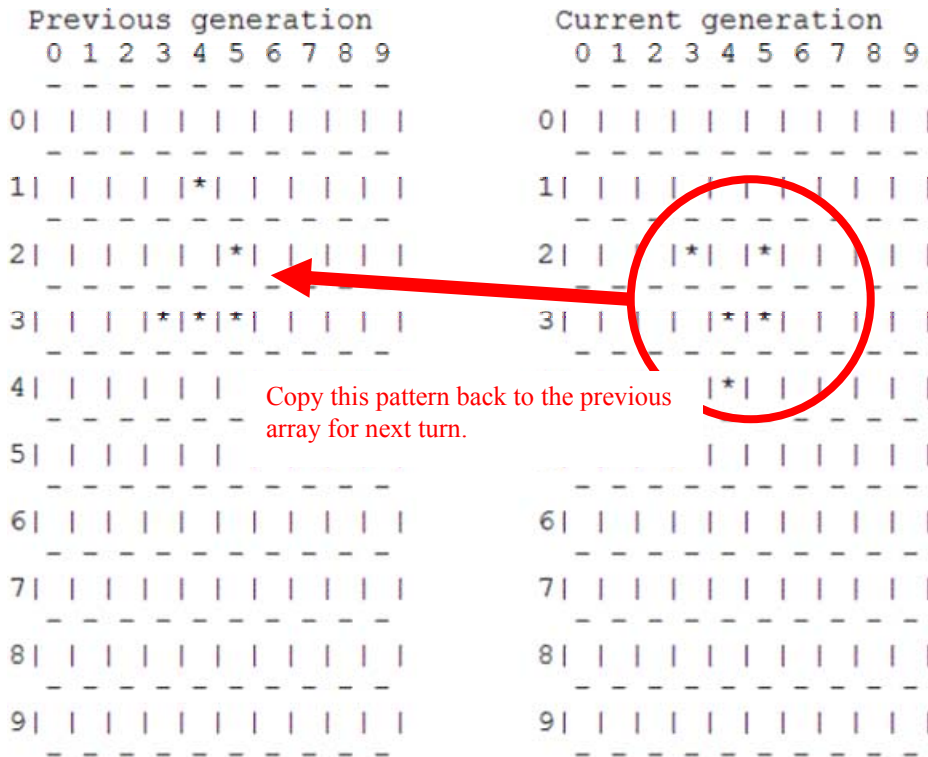
The copy method of class Biosphere is supposed to update the state of the previous array to the state of the current array after all the births and deaths have occurred. In the space provided below indicate what the logic error in this method is.


```
Previous generation              Current generation
  0 1 2 3 4 5 6 7 8 9              0 1 2 3 4 5 6 7 8 9
  - - - - - - - - - -              - - - - - - - - - -
0| | | | | | | | | | |          0| | | | | | | | | | |
  - - - - - - - - - -              - - - - - - - - - -
1| | | | | |*| | | | | |         1| | | | | | | | | | |
  - - - - - - - - - -              - - - - - - - - - -
2| | | | | |*| | | | |          2| | |  |*| |*| | |  | |
  - - - - - - - - - -              - - - - - - - - - -
3| | | |*|*|*| | | | |          3| | |  | |*|*| | |  | |
  - - - - - - - - - -              - - - - - - - - - -
4| | | | | | |                   |*| | | | | |
  - - - - - -                      - - - - - -
5| | | | | |                     | | | | | | | |
  - - - - - - - -                  - - - - - - - -
6| | | | | | | | | | | |         6| | | | | | | | | | |
  - - - - - - - - - -              - - - - - - - - - -
7| | | | | | | | | | | |         7| | | | | | | | | | |
  - - - - - - - - - -              - - - - - - - - - -
8| | | | | | | | | | | |         8| | | | | | | | | | |
  - - - - - - - - - -              - - - - - - - - - -
9| | | | | | | | | | | |         9| | | | | | | | | | |
  - - - - - - - - - -              - - - - - - - - - -
```

Copy this pattern back to the previous array for next turn.

Turn: 1

<< Begin answer space >>

The copy method is supposed to copy the pattern of stars and spaces from the previous array to the current array. Each array element contains a reference to a Critter so the assignment operator will cause each element in the previous array to point to the same Critter objects that each corresponding element in the current array refers to. Births and deaths are supposed to occur simultaneously in the Biosphere so you are to read the pattern of Critters from the previous array and make changes only in the current array. Because the references refer to the same Critters births and deaths will no longer occur at the same time.

<< End answer space >>

2. Describe the difference between the procedural and the object-oriented approaches to program design.

<< Begin answer space >>

With the procedural approach you are thinking of your program in terms of what different parts do and divide up your program into procedures which each perform a well defined task.

With the object-oriented approach you are thinking of your program in terms of the objects that it may consist of so you divide your program up by classes each of which consist of attributes and methods.

<< End answer space >>

3. For this question please refer to the following program:

```
class Driver
{
    public static void main (String [] args)
    {
        final int SIZE = 4;
        int i;
        Foo [] arr = new Foo[4];
```

<< Begin space to modify the Driver class >>

for (i = 0 ; i < SIZE; i++)
{
**   arr[i] = new Foo('i', i);**
}

for (i = 0; i < SIZE; i++)
{
**   System.out.println(arr[i].getCh() + " " + arr[i].getNum());**
}

<< End space to modify the Driver class >>

```
        }
    }
class Foo
{
    private char ch;
    private int num;

    public Foo ()
    {
        ch = 'a';
        num = 0;
    }
```

<< Begin space to modify class Foo >>

```
public Foo ()
{
    ch = 'a';
    num = 0;
}

public Foo (char c, int n)
{
    ch = c;
    num = n;
}

public void setCh (char c) { ch = c; }
public char getCh () { return ch; }
public void setNum (int n) { num = n; }
public int getNum () { return num; }
```

<< End space to modify class Foo >>

```
}
```

In the space provided above in the definitions for the Driver class and class Foo you are to:

1) Initialize the array elements so that the attributes of each instance of class Foo take on the following values:
   Element[0]'s instance of Foo: ch = 'i', num = 0
   Element[1]'s instance of Foo: ch = 'i', num = 1
   Element[2]'s instance of Foo: ch = 'i', num = 2
   Element[3]'s instance of Foo: ch = 'i', num = 3

2) Display the contents of the array so that they appear in the following format:
   i 0
   i 1
   i 2
   i 3

JT: Liked the review questions, then you'll love the midterm! ☻