

CS102, TEST 1, MONDAY, OCTOBER 5, 2009,
PROF. LOFTIN

SOLUTIONS

- (1) (6 pts) What is the output to the screen of the following program?

```
public class Question1{
    public static void main (String[] args){
        int[] ar = {3,6,10,4,12};
        System.out.println(ar[3]);
        change(ar);
        System.out.println(ar[3]);
    }
    public static void change (int[] a){
        for (int i=0; i<a.length; i++)
            a[i] += i;
    }
}
```

Solution:

4
7

- (2) (10 pts) Write a method

```
public static void skipPrint (int[] ar)
```

which prints to the screen every other element of the array `ar`, starting with the initial element. Make sure your code works whether `ar` has an even or an odd number of entries. For example, the array `{3,4,12,15,26,0,9}` is passed into `ar`, the output to the screen should be

3 12 26 9

Solution:

```
public static void skipPrint (int[] ar){
    for (int i=0; i<ar.length; i+=2)
        System.out.print (ar[i] + " ");
    System.out.println();
}
```

- (3) (10 pts) Put the classes `Mammal`, `Rodent`, `Dog`, `Rat`, `Cat`, `Lion`, `Leopard`, `Mouse`, `Whale`, `Cow` into a class hierarchy.

- (5) (6 pts) (In this problem, the class `Advice` was unintentionally omitted. Here is a corrected version of the problem, with a solution.)

Corrected version

Consider the following classes `Thought` and `Advice`:

```
public class Thought{
    public void message(){
        System.out.println("I feel like I'm diagonally parked in a " +
                           "parallel universe.");
    }
}

public class Advice extends Thought{
    public void message(){
        System.out.println("Warning: Dates in calendar are closer " +
                           "than they appear.");
        super.message();
    }
}
```

What is the output of the following driver class?

```
public class Messages{
    public static void main (String[] args){
        Thought a = new Thought();
        Thought b = new Advice();
        a.message();
        b.message();
    }
}
```

Solution:

I feel like I'm diagonally parked in a parallel universe.
Warning: Dates in calendar are closer than they appear.
I feel like I'm diagonally parked in a parallel universe.

- (6) (6 pts) What is the output to the screen of the following code?

```
int[] a = {3,6,7,1,12};
for (int val : a){
    if (val%2 == 0)
        System.out.print(val + " ");
    else
        System.out.print((val-1) + " ");
}
```

Solution:

2 6 6 0 12

- (7) (16 pts) Consider the following class Num, which is a user-defined wrapper class for an integer.

```
public class Num{
    private int n;
    public Num(int x){n=x;}
    public void setNum(int x){n=x;}
    public int getNum(){return n;}
    public String toString(){return n+"";}}
```

- (a) Write a method
`public static void switchNums(Num n1, Num n2)`
 which switches the values of the Num objects passed as parameters into n1 and n2.

Solution:

```
public static void switchNums(Num n1, Num n2){
    int temp = n1.getNum();
    n1.setNum(n2.getNum());
    n2.setNum(temp);
}
```

- (b) What is the output to the screen of the following driver class of Num?

```
public class NumDriver{
    public static void main(String[] args){
        Num a = new Num(5), b = new Num(7);
        System.out.println(a + " " + b);
        f(a,b);
        System.out.println(a + " " + b);
    }
    public static void f(Num c, Num d){
        c.setNum(10);
        d = new Num(3);
    }
}
```

Solution:

```
5 7
10 7
```

- (8) (6 pts) Consider the following two classes Parent and Child. There is an error in the following code. Fix it by circling the error and rewriting the correct code to the side.

```
public class Parent{
    private int n;
    public Parent(int num){ n=num; }
```

```
    public int getN(){ return n; }
}
public class Child extends Parent{
    public Child(int a){ super(a); }
    public double reciprocal(){ return 1./n;}
}
```

Solution: The error is that the `Child` class cannot access the `private` data `n`. One solution is to change `n` to be `protected` (so the second line should read

```
    protected int n;
```

Another solution (perhaps a better one from the point of view of encapsulation) is to rewrite the method definition of `reciprocal` to read

```
    public double reciprocal(){ return 1./getN();}
```

(9) (6 pts) What is the output to the screen?

```
public class NestedLoops{
    public static void main (String[] args){
        for (int i=1; i<=3; i++){
            for (int j=1; j<=4; j++)
                System.out.print("*");
            System.out.println();
        }
    }
}
```

Solution:

```
****
****
****
```