

CS102, TEST 2, MONDAY, NOVEMBER 9, 2009,
PROF. LOFTIN

SOLUTIONS

- (1) (4 pts) What is the output to the screen of the following program?

```
public class Question1{
    public static void main (String[] args){
        System.out.println(f(10));
    }
    public static int f(int n){
        if (n<2)
            return n;
        else
            return 1+f(n/2);
    }
}
```

Solution:

4

- (2) (4 pts) Inside a `paintComponent` (Graphics page) method, write the code to draw a black filled-in square of side length 40 whose upper left corner is at the point (30, 100).

Solution:

```
page.setColor (Color.black);
page.fillRect (30,100,40,40);
```

- (3) (10 pts) Consider F_n defined by

$$F_n = \begin{cases} 0 & \text{if } n = 0 \\ F_{n-1} + n(n-1) & \text{if } n > 0 \end{cases}$$

- (a) Write a recursive method which has the integer n as a parameter and which returns the integer F_n .

Solution:

```
public static int F(int n){
    if (n==0)
        return 0;
    else
```

```
        return F(n-1) + n*(n-1);
```

```
    }
```

- (b) Write an iterative method (i.e., a method using a loop) which also has parameter n and which returns F_n .

Solution:

```
public static int F(int n){
    int sum = 0;
    for (int i=1; i<=n; i++)
        sum += i*(i-1);
    return sum;
}
```

- (4) (6 pts) Consider the array given by declaration

```
int[] ar = {1,7,3,14,0,12};
```

Trace through the selection sort algorithm by writing out the elements of `ar` after each pass of the algorithm.

Solution:

```
1 7 3 14 0 12
0 7 3 14 1 12
0 1 3 14 7 12
0 1 3 14 7 12
0 1 3 7 14 12
0 1 3 7 12 14
```

- (5) (6 pts) What is the output to the screen?

```
public class Question5{
    public static void main (String[] args){
        try{ g(3); }
        catch (ArithmeticException e){
            System.out.println("Don't divide by zero.");
        }
        System.out.println("End of main.");
    }
    public static void g(int n){
        System.out.println(12/n);
        if (n<0) return;
        else g(n-1);
        System.out.println("End of g with n = " + n + ".");
    }
}
```

Solution:

4

```

6
12
Don't divide by zero.
End of main.

```

- (6) (8 pts) Consider the following method.

```

public static int h(int n){
    if (n==0) return 1;
    else return 2*h(n-1);
}

```

- (a) What is $h(5)$?

Solution: $h(5)$ is 32, since we may compute recursively

$$h(5) = 2h(4) = 2 \cdot 2h(3) = 2 \cdot 2 \cdot 2h(2) = 2 \cdot 2 \cdot 2 \cdot 2h(1) = 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2h(0) = 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 1 = 32$$

- (b) Find a formula for $h(n)$ for n any positive integer.

Solution: $h(n) = 2^n$. The same computations as above show that $h(n) = 2 \cdot 2 \cdots 2 \cdot 1$, where there are n 2's in the product. This is 2^n .

- (7) (10 pts) Consider the interface

```

public interface SolidShape{
    public double surfaceArea();
    public double volume();
}

```

A sphere of radius r has surface area $4\pi r^2$ and volume $\frac{4}{3}\pi r^3$. A cube of side length s has surface area $6s^2$ and volume s^3 . Write two classes `Sphere` and `Cube` which both implement the `SolidShape` interface. Include r as private data of the sphere and s as private data of the cube. Also write appropriate constructors. (Recall the constant π is given by `Math.PI` in Java.)

Solution:

```

public class Sphere implements SolidShape{
    private double r;
    public Sphere (double rad){
        r = rad;
    }
    public double surfaceArea(){
        return 4 * Math.PI * r*r;
    }
    public double volume(){
        return 4.0/3.0 * Math.PI * r*r*r;
    }
}

```

```
public class Cube implements SolidShape{
    private double s;
    public Cube (double side){
        s=side;
    }
    public double surfaceArea(){
        return 6 * s*s;
    }
    public double volume(){
        return s*s*s;
    }
}
```

- (8) (a) (3 pts) Write a class `SSNumberFormatException` which extends the `Exception` class and sets the exception message to "Invalid Social Security Number Format".

Solution:

```
public class SSNumberFormatException extends Exception{
    public SSNumberFormatException(){
        super("Invalid Social Security Number Format");
    }
}
```

- (b) (7 pts) Write a class `PersonalData` which has three `String` data fields representing the first name, last name and social-security number. Your class should:
- Have a constructor for the class to set the data.
 - Have an appropriate `toString()` method.
 - Implement the `Comparable` interface based on comparing social-security numbers.
 - The constructor should throw an `SSNumberFormatException` if the social-security number is not of the format `xxx-xx-xxxx`, where each `x` represents a digit from '0' to '9'.

Solution:

```
public class PersonalData implements Comparable{
    String fName, lName, ssNum;
    public PersonalData (String f, String l, String s)
        throws SSNumberFormatException{
        fName = f;
        lName = l;
        ssNum = s;
    }
}
```

```
        boolean valid = true;
        if (s.length() != 11)
            valid = false;
        else if (s.charAt(3) != '-' || s.charAt(6) != '-')
            valid = false;
        else
            for (int i=0; i<11; i++)
                if (i!=3 && i!=6 &&
                    (s.charAt(i)<'0' || s.charAt(i)>'9'))
                    valid = false;
        if (!valid)
            throw new SSNumberFormatException();
    }
    public int compareTo(Object other){
        return ssNum.compareTo( ((PersonalData)other).ssNum );
    }
    public String toString(){
        return ssNum + " " + lName + ", " + fName;
    }
}
```