

CS 101: Test 2: April 6, 2009

SOLUTIONS

- (10 pts) Mark each of the following logical expressions as either true or false. You can assume we have declared

```
int x=2, y=2, z=4;
boolean k = false;
```

Expression	true	false
<code>(x>=1) k</code>	X	
<code>(z==y*x)</code>	X	
<code>((z<5) (y*y*y<z))</code>	X	
<code>(!k && (y==2))</code>	X	
<code>(y==x++)</code>	X	

- (8 pts) What is the output of the following code fragment?

```
int a=3, x=2;
do{
    System.out.println(a+" "+x);
    if (a!=1) a--;
    x += a*a;
}while ((a!=0) && (x<9));
```

Solution:

Output: 3 2 2 6 1 7 1 8

- (6 pts) Write a method `isEven` which accepts one integer parameter and returns true if the parameter is even (and false if the parameter is odd).

Solution:

```
public static boolean isEven(int n){
    return n%2==0;
}
```

4. (15 pts) A cube with side length s has volume s^3 and surface area $6s^2$. Write a class `Cube` which contains the following public methods

```
Cube (double x)           // constructor for cube of side length x
double getSide()         // get side length
void setSide(double y)   // set side length to y
double area()            // returns surface area of cube
double volume()          // returns volume of cube
```

Your class should have whatever private data and methods are appropriate. Your constructor and `setSide` method should set the side length to 0 if an attempt is made to set it to a negative number.

Solution:

```
public class Cube{
    private double length;

    public Cube(double x){
        setSide(x);
    }

    public double getSide(){
        return length;
    }

    public void setSide(double y){
        if (y>0)
            length = y;
        else
            length = 0;
    }
}
```

```

public double area(){
    return 6*length*length;
}

public double volume(){
    return length*length*length;
}
}

```

5. (8 pts) Consider the following method:

```

public static String question5(String sentence){
    Scanner scan = new Scanner (sentence);
    String ret = scan.next();
    while (scan.hasNext()){
        String nextWord = scan.next();
        if (ret.toUpperCase().compareTo(nextWord.toUpperCase()) < 0)
            ret = nextWord;
    }
    return ret;
}

```

What is the return value of the above method, if the value passed into the parameter `sentence` is

"I slit a sheet; a sheet I slit; upon a slitted sheet I sit."

Explain your answer.

Solution: The return value is `upon`. This method returns the word in the sentence which is largest in terms of alphabetical order.

To analyze this code line by line, recall that `scan.next()` returns the next word (delimited by white space) in the relevant `String` being scanned. So, in order, the return values of `scan.next()` are given by the words `I`, `slit`, `a`, `sheet`;, `a`, `sheet`, `I`, `slit`;, `upon`, `a`, `slitted`, `sheet`, `I`, `sit`.

So the initial line `String ret = scan.next();` sets `ret` to "I". Upon entering the `while` loop, `nextWord` is set to the next word in the scanner "slit". Now the conditional in the `if` statement evaluates to `true` when `ret` comes before `nextWord` in alphabetical order (note that we've sent both of them to `toUpperCase()` before comparing them). Therefore, `ret` is set to "slit".

So in the next pass through the loop `nextWord` is "a", and `ret` retains the value "slit" since it comes after `nextWord` in alphabetical order.

In the next pass, `nextWord` is "sheet", and again `ret` retains the value "slit". Continuing in this way, the loop considers all the words in the scanner, and the one which comes last alphabetically is "upon". So "upon" will be the final value of `ret` and so is the return value of the method.

6. (12 pts)

(a) What is the returned value of the method `f`?

```
public static int f(int x, int y){return (x>y) ? y : x; }
```

- i. The sum `x + y`.
- ii. The difference `x - y`.
- iii. The maximum of `x` and `y` (i.e. the value of the larger of `x` and `y`).
- iv. The minimum of `x` and `y` (i.e. the value of the smaller of `x` and `y`).

(b) The following code fragment prints out _____ asterisks.

```
for (char c='P'; c > 'N'; c--) System.out.print('*');
```

- i. 0
- ii. 2
- iii. 3
- iv. 4

(c) Consider the code fragment What is the output?

```
boolean a=true;
for (int b=1; b<=3; b++){
    a = !a;
```

```

        if (a) System.out.print('+');
    }
    if (a) System.out.println('-');

    i. ++
    ii. +-
    iii. ++-
    iv. +

```

(d) Consider the following methods:

```

public static void spaces(int m){
    for (int i=1; i<=m; i++) System.out.print(" ");
}
public static void line(int m){
    for (int i=1; i<=m; i++) System.out.print("*");
    System.out.println();
}

```

Which code fragment prints out the following pattern?

```

*****
****
***
**
*

```

- i. `for (int i=1; i<=5; i++){ spaces(i-1); line(5-i); }`
- ii. `for (int i=1; i<=5; i++){ spaces(i-2); line(6-i); }`
- iii. `for (int i=1; i<=5; i++){ spaces(i-1); line(6-i); }`
- iv. `for (int i=1; i<=5; i++){ spaces(i); line(5-i); }`

7. (10 pts) Consider the class `BasketballTeam` on the last page (note this is EXACTLY the same as the class on the sample test). What is the output of the following driver program?

```
public class BigGame{
    public static void main(String[] args){
        BasketballTeam team1 = new BasketballTeam("Lions");
        BasketballTeam team2 = new BasketballTeam("Wildcats");
        team2.makeShot();
        team2.freeThrow();
        team1.threePoint();
        System.out.println ("Team 2 has " + team2.freeThrow() +
            " points.");
        for (int i=1; i<=4; i++){
            team2.makeShot();
            team1.threePoint();
        }
        team2.threePoint();
        System.out.print(team1);
        System.out.print(team2);
    }
}
```

```
Output:
Team 2 has 4 points.
The Lions have 15 points.
The Wildcats have 15 points.
```

```
public class BasketballTeam{
    private int score;
    private String name;

    public BasketballTeam(String s){
        score = 0;
        name = s;
    }

    public int getScore(){
        return score;
    }

    public String getName(){
        return name;
    }

    public String toString(){
        return ("The " + name + " have " + score + " points.\n");
    }

    public int freeThrow(){
        score++;
        return score;
    }

    public int threePoint(){
        score += 3;
        return score;
    }

    public int makeShot(){
        score += 2;
        return score;
    }
}
```