

### Homework Assignment 3 Solution

1. Let the variables a, b, and c be of type boolean and a and c have been assigned a value. What value is assigned to b in each of these four assignment statements? Show your work for full credit.

(a) `b = (8 < 10 && false);`  
`b = (true && false);`  
`b = false;`

(b) `b = (Math.pow(2.0, 3.0) > Math.ceil(7.2));`  
`b = (8.0 > 8.0);`  
`b = false;`

(c) `b = !true || (!false && true);`  
`b = false || (true && true);`  
`b = false || true;`  
`b = true;`

(d) `b = (a || !c) == (c && !a);` Hint: think truth table  
Truth table:

a	c	a    !c	c && !a	b
T	T	T	F	F
T	F	T	F	F
F	T	F	T	F
F	F	T	F	F

`b = false`

2. Let the variables x, y, and z, be of type int and assigned some values. The variable b1 is initialized as follows:

```
boolean b1 = (x > z) && (y >= z);
```

For each boolean value b2, b3, b4, and b5 defined below, state whether it ALWAYS, SOMETIMES, or NEVER has the same value as b1. If it can have a different value from b1, give an example of values x, y, z for which it is different.

(a) `boolean b2 = (x <= z) || (y < z)`  
`b2 = !(x <= z) || (y < z)`  
`b2 = !(!(x <= z) && !(y < z))`

`b2 = !((x > z) && (y >= z))`  
b2 NEVER has the same value as b1.

(b) boolean  $b3 = !((x < z) \ || \ (y \leq z))$   
 $b3 = !(x < z) \ \&\& \ !(y \leq z)$   
 $b3 = (x \geq z) \ \&\& \ (y > z)$

$b1 \neq b2$  when  $(x > z) \ \&\& \ (y == z)$  or  $(x == z) \ \&\& \ (y > z)$   
 eg. if  $x = 6, y = 5, z = 5$ , then  $b1 = \text{true}$  and  $b3 = \text{false}$

otherwise  $b1 == b2$   
 eg. if  $x = 6, y = 5, z = 4$ , then  $b1 = b3 = \text{true}$

$b3$  SOMETIMES has the same value as  $b1$ .

(c) boolean  $b4 = !((x < z) \ \&\& \ (y \geq z))$   
 $b4 = !(x < z) \ || \ !(y \geq z)$   
 $b4 = (x \geq z) \ || \ (y < z)$

if  $x = 6, y = 5, z = 4$ , then  $b1 = \text{true}$  and  $b4 = \text{true}$   
 if  $x = 2, y = 4, z = 5$ , then  $b1 = \text{false}$  and  $b4 = \text{true}$

$b4$  SOMETIMES has the same value as  $b1$ .

(d) boolean  $b5 = ((x > z) \ \&\& \ (y > z)) \ || \ ((x > z) \ \&\& \ (y == z))$

Truth table

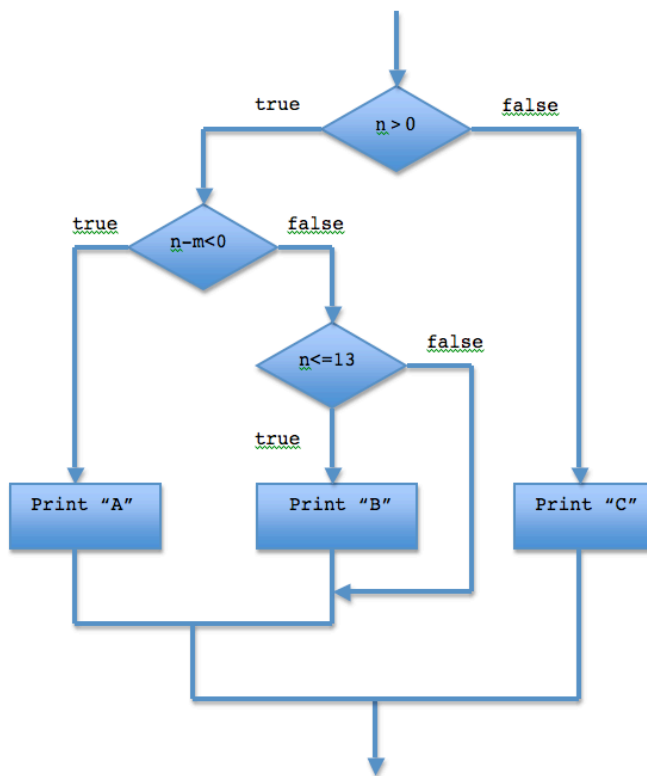
$x > z$	$(y > z)$	$(y == z)$	$((x > z) \ \&\& \ (y > z))$	$((x > z) \ \&\& \ (y > z)) \    \ ((x > z) \ \&\& \ (y == z))$
T	T	F	T	$T \    \ F = T$
T	F	T	T	$F \    \ T = T$
T	F	F	F	$F \    \ F = F$
F	T	F	F	$F \    \ F = F$
F	F	T	F	$F \    \ F = F$
F	F	F	F	$F \    \ F = F$

$b5$  ALWAYS has the same value as  $b1$

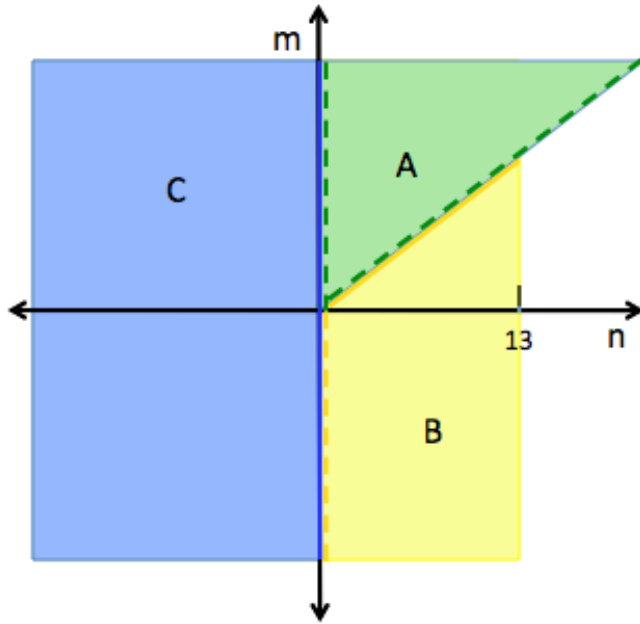
3. In the following Java code fragment, assume that the int variables n and m have already been initialized:

```
if (n > 0) {  
    if (n - m < 0) {  
        System.out.print("A");  
    }  
    else if (n <= 13) {  
        System.out.print("B");  
    }  
}  
else {  
    System.out.print("C");  
}
```

(a) Draw a flowchart to illustrate the flow of execution of the fragment. (You may write "print A", "print B", and "print C", for the print statements.)



(b) Draw a graph like the one shown below, shading in the area where "A" would be output in one color, the area where "B" would be output in another color, and the area where "C" would be output in a third color. Any other area should be left uncolored. If the border of an area is included in the shaded area, draw the border as a solid line. Otherwise, draw the border as a dashed line.



4. A sports bar will give a customer a free taco appetizer if all of the following conditions apply:

- \* The customer brings a "terrible towel".
- \* The customer is wearing a Steelers jersey.

(a) Complete the following Java statement that prints whether the taco appetize is free or not, where attire is a String variable and hasTerribleTowel is a boolean variable that have been initialized.

```
if ((attire.equals ("Steelers jersey")) && hasTerribleTowel == true ) {
    System.out.println("Free");
}
else {
    System.out.println("$3.50");
}
```

(b) Use de Morgan's law to rewrite the if statement in part (a) without using the same logical operator you used in part (a).

Solution: replace the if (condition) with if (!(condition)) and then apply de Morgan's law to the inner !:

```
if (!( !(attire.equals ("Steelers jersey"))
      || hasTerribleTowel == false )){
    System.out.println("Free");
}
else {
    System.out.println("$3.50");
}
```

(c) Rewrite the if statement in part (a) without using the keywords true or false.

```
if ( attire.equals("Steelers jersey") && hasTerribleTowel )
```

## PART II

```
/*
*****
• Problem 5
*****
/*
* Determines whether the integer the user enters is a perfect
* square or not.
*
*/

import java.util.Scanner;
public class PerfectSquareTester {

    public static void main(String[] args){

        Scanner sc = new Scanner(System.in);
        System.out.println("please input an integer");
        int num = sc.nextInt();

        boolean isPerfectSquare;
        if(num >= 0 {
            // truncate sqrt(num) to an integer
            // eg, sqrt(15)=>3, sqrt(16)=>4, sqrt(17)=>4
            int result = (int) Math.sqrt(num);

            // square the result to see if get original number
            result = result * result;

            if (result == num) {
                isPerfectSquare = true;
            }
            else {
                isPerfectSquare = false;
            }
            // alternately you could write:
            // isPerfectSquare = (result == num);
        }
        else { // negative numbers are not perfect squares
            isPerfectSquare = false;
        }
        System.out.println(isPerfectSquare);
    }
}
```

```

/*****
 * Problem 6
 *****/
 *
 * Computes the cost of three-day two-night weekend stay at a golf resort
 * for three package types: room only, room plus meals, and "stay and play".
 * The cost of a package depends on the number of persons sharing the room
 * and the number of rounds of golf.
 */

import java.util.Scanner;

public class GolfResortCalculator {

    public static void main(String[] args) {
        // SET-UP
        Scanner in = new Scanner(System.in);
        displayMenu();
        String choice=in.nextLine();

        if (choice.equals("RO")){
            // ROOM ONLY
            System.out.println();
            displayRoomOnlyCost();
        }
        else if (choice.equals("RM")){
            // ROOM PLUS MEALS
            System.out.print("Enter the number of persons staying in the room: ");
            int numPersons=in.nextInt();
            System.out.println();
            displayRoomPlusMealsCost(numPersons);
        }
        else if (choice.equals("SP")){
            // STAY AND PLAY
            System.out.print("Enter the number of persons staying in the room: ");
            int numPersons=in.nextInt();
            System.out.print("Enter the number of rounds of golf: ");
            int numRounds=in.nextInt();
            System.out.println();
            displayStayAndPlayCost(numPersons,numRounds);
        }
        else {
            // INVALID CHOICE
            System.out.println();
            System.out.println("ERROR: This package is unknown");
        }
    }

    // Displays the main menu
    public static void displayMenu(){
        out.println("Golf Package Menu:");
        out.println("RO [Room only]");
        out.println("RM [Room plus meals]");
        out.println("SP [Stay and play]");
        out.println();
        out.print("Enter the menu choice: ");
    }
}

```

```

// Displays cost for room only package
public static void displayRoomOnlyCost(){
    int initialCents = 26495;

    displayCost(initialCents);
}

// Displays cost of room plus meal package given number of persons
public static void displayRoomPlusMealsCost(int numPersons){
    int initialCents = 24995;
    int centsPerPerson = 14550;
    int finalCents = initialCents + numPersons * centsPerPerson;

    displayCost(finalCents);
}

// Displays cost of stay and play package given number of persons and
// rounds of golf
public static void displayStayAndPlayCost(int numPersons, int numRounds){
    int initialCents = 22495;
    int centsPerPerson = 13525;
    int centsPerRoundFirstTwo = 11540;
    int centsPerRoundAfterTwo = 9540;

    int finalCents = initialCents + numPersons * centsPerPerson;

    if (numRounds > 2) {
        // add cost for rounds after first two
        finalCents += (numRounds - 2) * centsPerRoundAfterTwo;
        numRounds = 2;
    }
    // add cost for up to two rounds
    finalCents += numRounds * centsPerRoundFirstTwo;

    displayCost(finalCents);
}

// Displays cost in dollars and cents, given cost in cents
public static void displayCost(int numCents){
    int dollars = numCents / 100;
    int cents = numCents % 100;

    String cost = "$" + dollars + ".";
    if (cents < 10) cost += "0"; // append 0
    cost += cents; // append cents
    System.out.println("The package cost is " + cost);
}
}
}

```