

Home Assignment 3

CPE 102

Chapter 5

1. In some Asian countries, the number 14 is considered unlucky. Some building owners play it safe and skip both the thirteenth and the fourteenth floor. How would you modify the sample program to handle such a building?

Subtract two from floors greater than 14 to remove both the bad floors.

2. Consider the following if statement to compute a discounted price:

```
if (originalPrice > 100)
{
    discountedPrice = originalPrice - 20;
}
else
{
    discountedPrice = originalPrice - 10;
}
```

What is the discounted price if the original price is 95? 100? 105?

85, 90, 85,

3. Compare this if statement with the one in Self Check 2:

```
if (originalPrice < 100)
{
    discountedPrice = originalPrice - 10;
}
else
{
    discountedPrice = originalPrice - 20;
}
```

Do the two statements always compute the same value? If not, when do the values differ?

No, because they are not inclusive if statements. In one the value of 100 will be 90 and the other 80.

4. Consider the following statements to compute a discounted price:

```
discountedPrice = originalPrice;  
if (originalPrice > 100)  
{  
    discountedPrice = originalPrice - 10;  
}
```

What is the discounted price if the original price is 95? 100? 105?

95, 100, 95,

5. The variables fuelAmount and fuelCapacity hold the actual amount of fuel and the size of the fuel tank of a vehicle. If less than 10 percent is remaining in the tank, a status light should show a red color; otherwise it shows a green color. Simulate this process by printing out either "red" or "green".

```
If fuelAmount < fuelCapacity*0.1  
    Print red  
else print green
```

6. Which of the following conditions are true, provided a is 3 and b is 4?

- a. $a + 1 \leq b$
 - This is true $4=4$
- b. $a + 1 \geq b$
 - True, $4=4$
- c. $a + 1 \neq b$
 - False

7. Give the opposite of the condition

```
floor > 13  
• Floor  $\leq 13$ 
```

8. What is the error in this statement?

```
if (scoreA = scoreB)  
{  
    System.out.println("Tie");  
}
```

- scoreA is assigned to scoreB not compared with ==

9. Supply a condition in this if statement to test whether the user entered a Y:

```
System.out.println("Enter Y to quit.");  
String input = in.next();  
if ( . . . )
```

```
{  
    System.out.println("Goodbye.");  
}
```

- `input.equals("Y")`

10. Give two ways of testing that a string `str` is the empty string.

- `Str.equals("")` or `"".equals(str)`

11. What is the value of `s.length()` if `s` is

- the empty string `""`?
 - 0
- the string `" "` containing a space?
 - 1
- `null`?
 - `NullPointerException`

12. Which of the following comparisons are syntactically incorrect? Which of them are syntactically correct, but logically questionable?

```
String a = "1";  
String b = "one";  
double x = 1;  
double y = 3 * (1.0 / 3);
```

- `a == "1"`
 - `a == null`
 - `a.equals("")`
 - `a == b`
 - `a == x`
 - `x == y`
 - `x - y == null`
 - `x.equals(y)`
- Questionable: A, D, F,
 - Incorrect: E, G, H,

13. In a game program, the scores of players A and B are stored in variables `scoreA` and `scoreB`. Assuming that the player with the larger score wins, write an if/else if/else sequence that prints out "A won", "B won", or "Game tied".

```
if (scoreB > scoreA)  
{
```

```
System.out.println("B won");  
}  
else if (scoreB < scoreA)  
{  
System.out.println("A won");  
}  
else  
{  
System.out.println("Game tied");  
}
```

14. Write a conditional statement with three branches that sets *s* to 1 if *x* is positive, to -1 if *x* is negative, and to 0 if *x* is zero.

```
if (x>0) { s=1; }  
else if (x<0){ s=-1; }  
else { s=0; }
```

15. How could you achieve the task of Self Check 14 with only two branches?

```
s=1;  
if ( x=0 ) { s=0; }  
else if ( x<0 ) { s=-1; }
```

or

```
if( x=0 ){ s=0; }  
( s>0 ) ? 1:-1;
```

16. Beginners sometimes write statements such as the following:

```
if (price > 100)  
{  
discountedPrice = price - 20;  
}  
else if (price <= 100)  
{  
discountedPrice = price - 10;  
}
```

Explain how this code can be improved.

- No need for the else if evaluation because it will always be true if the first is not. Drop the if conditional.

17. Suppose the user enters -1 into the earthquake program. What is printed?

- "No destruction"

18. Suppose we want to have the earthquake program check whether the user entered a negative number. What branch would you add to the if statement, and where?

```
else if (richter < 0) { System.out.println("Error: Negative input"); }
```

- Functionally anywhere after the first if statement would work but ideally it would be the last one before the else as it would hopefully not be evaluated often.

19. What is the amount of tax that a single taxpayer pays on an income of \$32,000?

- 10% of 32000= \$3200

20. Would that amount change if the first nested if statement changed from

```
if (income <= RATE1_SINGLE_LIMIT)
to
if (income < RATE1_SINGLE_LIMIT)
```

- No because the next calculation would just cancel this out.

21. Suppose Harry and Sally each make \$40,000 per year. Would they save taxes if they married?

- No they wouldn't. Tax burden would be the same.

22. How would you modify the TaxCalculator.java program in order to check that the user entered a correct value for the marital status (i.e., Y or N)?

- Rather than using the else catchall use an else if instead which verifies that "N" was entered. Then use the else to print an error message.

23. Some people object to higher tax rates for higher incomes, claiming that you might end up with less money after taxes when you get a raise for working hard. What is the flaw in this argument?

- You are only taxed at the higher rate for income earned at that level. Thus the brackets.

24. Draw a flowchart for a program that reads a value temp and prints "Frozen" if it is less than zero.

25. What is wrong with the flowchart at right?

- Multiple conditionals use the same result.

26. How do you fix the flowchart of Self Check 25?

- Create separate results for each conditional.

27. Draw a flowchart for a program that reads a value x . If it is less than zero, print "Error". Otherwise, print its square root.

28. Draw a flowchart for a program that reads a value $temp$. If it is less than zero, print "Ice". If it is greater than 100, print "Steam". Otherwise, print "Liquid".

29. Using Figure 1 on page 181 as a guide, follow the process described in this section to design a set of test cases for the ElevatorSimulation.java program in Section 5.1.

Test Case	Expected	Why
12	12	No change, below 13
14	13	Above 13
13	NA	Unclear

30. What is a boundary test case for the algorithm in How To 5.1 on page 193? What is the expected output?

- The switching points between the two price points would be such a boundary. The dollar amount \$128.

31. Using Figure 4 on page 197 as a guide, follow the process described in Section 5.6 to design a set of test cases for the Earthquake.java program in Section 5.3.

Test Value	Expected	Why
8.5	Most structures	
8.0	Most structures	
7.5	Many buildings	
6.5	Many	
5	Damage to property	
3	No damage	
-1	Error	Corrected handling later

32. Suppose you are designing a part of a program for a medical robot that has a sensor returning an x- and y-location (measured in cm). You need to check whether the sensor location is inside the circle, outside the circle, or on the boundary (specifically, having a distance of less than 1 mm from the boundary). Assume the circle has center (0, 0) and a radius of 2 cm. Give a set of test cases.

Test Value	Expected	Why
(0,0)	Inside	
(5,5)	Outside	
(2,0)	On boundary	Exact
(0,1.99)	On boundary	Within .01
(0,1.9)	Inside	Close
(2.2,0)	Outside	Close

33. Suppose x and y are two integers. How do you test whether both of them are zero?

`X == 0 && Y == 0`

34. How do you test whether at least one of them is zero?

`X == 0 || Y == 0`

35. How do you test whether exactly one of them is zero?

`(X == 0 != Y == 0)`

36. What is the value of `!!frozen`?

`frozen`

37. What is the advantage of using the type boolean rather than strings "false"/"true" or integers 0/1?

- Booleans only have two values. Also can be used with other math where strings cannot.

38. In the ElevatorSimulation2 program, what is the output when the input is

- a. 100?
 - i. Error: must be between 1-20
- b. -1?
 - i. Error: must be between 1-20
- c. 20?
 - i. 19
- d. thirteen?
 - i. Not an int

39. Your task is to rewrite lines 19–26 of the ElevatorSimulation2 program so that there is a single if statement with a complex condition. What is the condition?

```
if ( floor < 0 || floor == 13 || floor > 20)
{
    System.out.println("Error: Invalid floor number");
}
```

40. In the Sherlock Holmes story “The Adventure of the Sussex Vampire”, the inimitable detective uttered these words: “Matilda Briggs was not the name of a young woman, Watson, ... It was a ship which is associated with the giant rat of Sumatra, a story for which the world is not yet prepared.” Over a hundred years later, researchers found giant rats in Western New Guinea, another part of Indonesia. Suppose you are charged with writing a program that processes rat weights. It contains the statements

```
System.out.print("Enter weight in kg: ");
double weight = in.nextDouble();
```

What input checks should you supply? When processing inputs, you want to reject values that are too large. But how large is too large? These giant rats, found in Western New Guinea, are about five times the size of a city rat.

```
if (in.hasNextDouble()) {
    double weight = in.nextDouble();
    if (weight < 0) {
        System.out.println("Error: Negative");
    }
    else if (weight > 100) {
        System.out.println("Error: Large Value");
    }
    else {
        <blah>
    }
}
else{
```



```
System.out.print("Error: NaN");  
}
```

41. Run the following test program and supply inputs 2 and three at the prompts. What happens? Why?

```
import java.util.Scanner  
public class Test  
{  
    public static void main(String[] args)  
    {  
        Scanner in = new Scanner(System.in);  
        System.out.print("Enter an integer: ");  
        int m = in.nextInt();  
        System.out.print("Enter another integer: ");  
        int n = in.nextInt();  
        System.out.println(m + " " + n);  
    }  
}
```

- It doesn't compile because import needs a semi-colon.

Chapter 7

1. Declare an array of integers containing the first five prime numbers.

- `int[] prime = { 2, 3, 5, 7, 11};`

2. Assume the array `primes` has been initialized as described in Self Check 1. What does it contain after executing the following loop?

```
for (int i = 0; i < 2; i++)  
{  
    primes[4 - i] = primes[i];  
}
```

- 2, 3, 5, 3, 2

3. Assume the array `primes` has been initialized as described in Self Check 1. What does it contain after executing the following loop?

```
for (int i = 0; i < 5; i++)  
{  
    primes[i]++;  
}
```

- 3, 4, 6, 8, 12,

4. Given the declaration

```
int[] values = new int[10];
```

write statements to put the integer 10 into the elements of the array values with the lowest and the highest valid index.

- Values[0] = 10;
- Values[values.length -1] =10;

5. Declare an array called words that can hold ten elements of type String.

- String[] words = new String[10];

6. Declare an array containing two strings, "Yes", and "No".

- String[] yn = { "Yes", "No" };

7. Can you produce the output on page 312 without storing the inputs in an array, by using an algorithm similar to the algorithm for finding the maximum in Section 6.7.5?

- No, because without storing the only way to "save" a value is to print it and you cannot retroactively add the max value array.

8. Declare a method of a class Lottery that returns a combination of n numbers. You don't need to implement the method.

```
public class Lottery
{
    public int[] getCombo(int n) { }
```

9. What does this enhanced for loop do?

```
int counter = 0;
for (double element : values)
{
    if (element == 0) { counter++; }
}
```

- It iterates through values where element is the value of the current assignment. When an element is equal to 0, it counts it.

10. Write an enhanced for loop that prints all elements in the array values.

```
For(int currentValue : values)
{
    System.out.println(currentValue);
}
```

11. Write an enhanced for loop that multiplies all elements in a double[] array named factors, accumulating the result in a variable named product.

```
For (double currentFactor : factor)
{
    Double Product *= factor;
}
```

12. Why is the enhanced for loop not an appropriate shortcut for the following basic for loop?

```
for (int i = 0; i < values.length; i++) { values[i] = i * i; }
```

- There is no i index in an enhanced for.

13. Given these inputs, what is the output of the LargestInArray program?

```
20 10 20 Q
20 <= Largest
10
20 <= Largest
```

14. Write a loop that counts how many elements in an array are equal to zero.

```
int count = 0;
for (double currentIndex : array)
{
    if (currentIndex == 0) { count++; }
}
```

15. Consider the algorithm to find the largest element in an array. Why don't we initialize largest and i with zero, like this?

```
double largest = 0;
for (int i = 0; i < values.length; i++)
{
    if (values[i] > largest)
    {
        largest = values[i];
    }
}
```

- Because the values could be negative.

16. When printing separators, we skipped the separator before the initial element. Rewrite the loop so that the separator is printed after each element, except for the last element.

```
for (int i = 0; i < values.length; i++)
{
    System.out.print(values[i]);
}
```

```
if (i < values.length - 1)
{
    System.out.print(" | ");
}
}
```

17. What is wrong with these statements for printing an array with separators?

```
System.out.print(values[0]);
for (int i = 1; i < values.length; i++)
{
    System.out.print(", " + values[i]);
}
```

- The array could have nothing in it.

18. When finding the position of a match, we used a while loop, not a for loop. What is wrong with using this loop instead?

```
for (pos = 0; pos < values.length && !found; pos++)
{
    if (values[pos] > 100)
    {
        found = true;
    }
}
```

- Pos will be one too high after a match is found.

19. When inserting an element into an array, we moved the elements with larger index values, starting at the end of the array. Why is it wrong to start at the insertion location, like this?

```
for (int i = pos; i < currentSize - 1; i++)
{
    values[i + 1] = values[i];
}
```

- You're setting each array element equal to pos.