

Group E
Teresa Luz Miller
Ahmed Elhendy
Katie Moyle

5.1. The Java program will still print the lowest value of the two or more.

The printed result is determined in whatever case. Let's say the two equal values are lower than the third, then only one of the equal but lower values will be printed. On the flip side, if the two equal values are higher than the third value, the third value will be printed.

5.2. The error with the follow code fragment is that is missing opening and closing brackets ({}) and before the if and else statement.

5.3. The symbol (=), which is termed the assignment operator, is used incorrectly. There should be two operators instead of one, as there's a difference between equals to and the true statement.

5.4 An error is thrown. We are calling a method that accepts a float but we are passing it as a double. The value must be clearly converted into a float in order for the program to run successfully.

5.5. The three output statements are printed: →

One is printed!
Two is printed!
Three is printed!

5.6. The list of strings in a lexicographic order:

```
"      "  
"*****"  
"12345"  
"6789"  
",+<?"  
"?-?-?-?"  
"Ethel"  
"HEPHALUMP"  
"Lucy"  
"AAAAAAAAAAAA"  
"Book"  
"Bookkeeper"  
"Fred"  
"Hephalump"  
"Ricky"
```

"{()}"

5.7. Output: →

900
800
700
600
500
400
300
200
100

5.8. Output: →

1
2
4
8
16
32
64
128

5.9. There are three distinct ways the program can be changed to remove the flaw of the infinite loop - because the count starts off initially greater than zero and then continues to increase in the value. The first way it can be fixed is having a decrementing in count instead of an incrementing count, the second way is that the count can initialize to 0 and using the while loop, count <= 50. The third way is choosing a limit and using the while loop, count less than or equal to Limit.

5.10. While loop:

```
System.out.print("Please enter a positive integer: ");  
Positive = scan.nextInt();
```

```
    While (positive <=0)  
{  
    system.out.print("Please enter a POSITIVE integer:");  
    Positive = scan.nextInt();  
}
```

5.11.

```
System.out.print ("Enter an integer (" + SENTINEL + " to quit): ");  
usernum = scan.nextInt();  
while (usernum != SENTINEL)
```

```
{
    System.out.println (usernum);
    usernum= scan.nextInt();
}
```

```
5.12. Public static boolean reminder(int yes, int no){
    If(no != 0){
        Int totalNum= yes/no;
    }
    Return totalNum;
}
```

```
5.13. Public boolean largerSum(double num1, double num2, double num3){
    if (num1 + num2 < num3){
        Return true;
    }else{
        Return false;
    }
}
```

```
5.14. Public boolean complexEquation(int prod1, int prod2, int prod3){
    if(prod1 != prod2) ||
        (prod1 + prod2 != prod3)){
        Return true ;
    }else{
        Return false;
    }
}
```

```
5.15. Public static boolean isAlpha(char d){
    if((d >= 'A' && d <= 'Z') ||
        (d >= 'a' && d <= 'z')) {
        Return true;
    }
    Return false;
}
```

```
5.16. public static boolean floatEquals(double fpv1, double fpv2, double tolerance)
{
    return(Math.abs(fpv1 -fpv2) <= tolerance);
}
```

```

5.17. Public static boolean isIsosceles(int sideOne, int sideTwo, int sideThree){
    if((sideOne == sideTwo && sideTwo != sideThree) ||
        ((sideOne != sideTwo && sideTwo == sideThree) ||
        ((sideOne != sideTwo && sideOne == sideThree)) {
        Return true;
    }
    Return false;
}
}
}

```

5.18. There are three radio buttons that are used in the program "QuoteOptions" that represent exclusive choices, which are comedy, philosophy or carpentry. They are all connected to each other through the ButtonGroup object, which allows only one of them from being selected. The meaning behind each choice associating them to JRadioButtons. These buttons are then added to a JPanel. If the buttons weren't organized to a ButtonGroup then multiple buttons could be selected at the same time but only the last selected button would show the quote referenced to it.