ACCESSIBILITY CHECKER

- File > Info > Check for Issues > Check Accessibility

STYLES

Verifying your Styles

- View Menu/Show/Navigation Pane

Modifying Font Size in Styles

- Verify all font sizes for Styles are a minimum size
- This will change any new document you create on your pc
- Where to find it
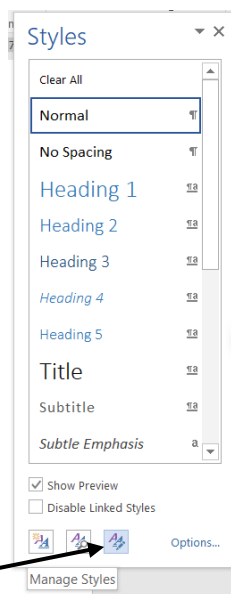 - Home > Styles pane > Manage Styles > Set Defaults, click bottom right corner and modify point size
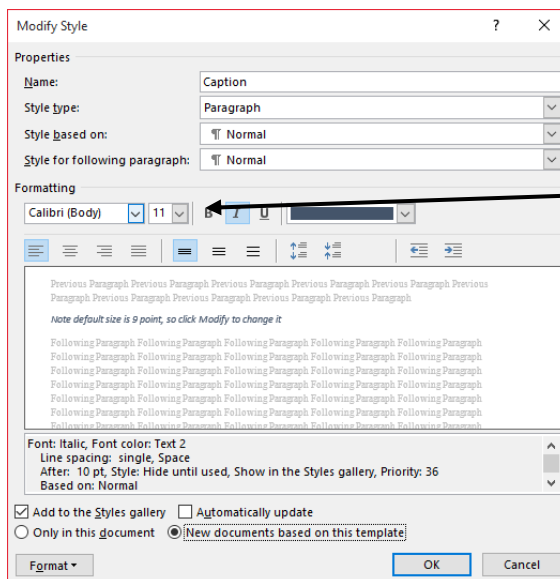

Figure 2 Manage Styles Window

Figure 1 Modify Style Window

CREATING TABLES

Alt Text

- Select the table
- Right Click on table and select Table Properties

Identify Header Row:

- Select the header row
- Go to Table Tools Layout
- Select Repeat Header Row

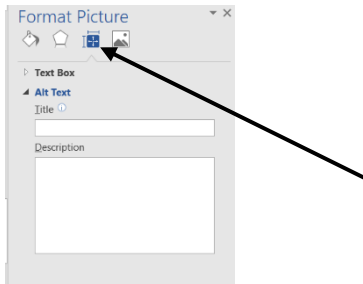| Class	Topic	Number of Students
Java Programming	Intro to Programming	112

IMAGES

Put Images In-line with Text

- In this example, the screen read will not recognize this image.

Adding Alt-Text to Images

- Right-click on Image, and select Format Picture then Layout and Properties

Captions

- Right Click on Picture and insert a caption

INCREASING FONT SIZES ISSUES

- Verify what your material looks like if you need to increase the font

23. What would you insert where you see the *********

```
for (int i = 0; i < *********; i++) {
    num = rnd.nextInt(max + 1) ;
    if (num < lowest) {
        lowest = num;
    } else if (num > highest) {
        highest = num;
    }
}
```

Problems with Multiple Columns

Course Objectives

1. Use the concepts of object-oriented programming to create Java programs that solve a variety of problems. [1, 2, 6a, 6b]
2. Apply fundamental UML techniques to the design of object-oriented programs. [1, 2, 6a, 6b]
3. Incorporate the concepts of inheritance and polymorphism in the design of Java classes. [1, 2, 6a, 6b]
4. Design Graphical User Interfaces (GUIs) using AWT/Swing or JavaFX. [1, 2, 6a, 6b]
5. Incorporate the use of String and array objects in the design of Java classes. [1, 2, 6a, 6b]
6. Incorporate the use of conditions, loops, and recursions in the design of object-oriented programs. [1, 2, 6a, 6b]

Program Outcomes

1. Analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions.
2. Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline.
6. Options for CS and IT:
   a) Apply computer science theory and software development fundamentals to produce computing-based solutions. [CS]
   b) Identify and analyze user needs and to take them into account in the selection, creation, integration, evaluation, and administration of computing-based systems. [IT]