Complete the following exercises. Take screenshots after each exercise and insert them into this Word document (make your screenshots large enough to read—expand the space after each heading). I recommend you use Snipping Tool to take your screenshots.

After you have completed the above exercises and taken screenshots, answer the questions.

Exercise 1.2: What happens if you call moveDown() twice? Three times?

A screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generated

The object gets closer and closer to the bottom of the canvas.

What happens if you call makeInvisible() twice?

The circle disappears and does not reappear after the second call.

Exercise 1.6: Invoke the changeColor() method. Write the color into the parameter field *without* the quotes. What happens?

An error message shows: “cannot find symbol – variable [yellow]”

Exercise 1.7: Complete and include a screenshot.

A close up of a logo

Description automatically generated

Exercise 1.8: Complete and include a screenshot.

Before moveRight() call:

**A screenshot of a cell phone

Description automatically generated**

After:

**A screenshot of a cell phone

Description automatically generated**

Exercise 1.9: Recreate both images and include a screenshot.

A picture containing game

Description automatically generatedA screenshot of a cell phone

Description automatically generated

Exercise 1.10: Complete and include a screenshot.

**A screenshot of a cell phone

Description automatically generated**

Exercise 1.14: How do you think the picture class draws the picture?

It adds all the shapes first, changes their color and size, then moves each shape to their location.

Exercise 1.16: Complete and include a screenshot.

**A screenshot of a cell phone

Description automatically generated**

Exercise 1.17: Complete and include a screenshot.

**A screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generatedA picture containing graphics, game

Description automatically generated**

Exercise 1.31: What are the types of the values listed in the book?

int, String, int, int, boolean, string, double

Exercise 1.32: How would you go about adding a new field (like **name**) to a circle object?

Navigate to the circle class and add “private [String] name”

Exercise 1.33: Write the header for a method named **send** that has one parameter of type **String**, and does not return a value.

*public void send(String message){}*

Exercise 1.34: Write the header for a method named **average** that has two parameters, both of type **int**, and returns an **int** value.

*public int average(int numOne, int numTwo){}*

Exercise 1.35: Is the homework you are currently working on an object or a class? If it is a class, name some of its objects. If it is an object, name its class.

It is a class, object examples can be circle1 and circle2

Exercise 1.36: Can an object have several different classes? Discuss. Keep your answer concise.

Yes, “inheritance” allows objects to use calls from other [parent] classes.

The following exercises are taken from pages 44-45 in the BlueJ book.

Exercise 1.23: Create an object of class **LabClass**. As the signature indicates, you need to specify the maximum number of students in that class (an integer)

A screenshot of a cell phone

Description automatically generated

Exercise 1.24: Call the **numberOfStudents** method of that class. What does it do? **See question 1**.

It returns 0 when called:

A screenshot of a cell phone

Description automatically generated

Exercise 1.25: Look at the signature of the **enrollStudent** method. You will notice that the type of the expected parameter is **Student**. Make sure you have two or three students and a **LabClass** object on the object bench, then call the **EnrollStudent** method of the **LabClass** object. With the input cursor in the dialog entry field, click on one of the student objects; this enters the name of the student object into the parameter field of the **enrollStudent** method. Click OK and you have added the student to the **LabClass**. Add one or more other students as well.

A screenshot of a cell phone

Description automatically generated

Exercise 1.26: Call the **printList** method of the **LabClass** object. You will see a list of all the students in that class printed to the BlueJ terminal window.

A screenshot of a cell phone

Description automatically generated

Exercise 1.27: Create three students with the following details:

* *Snow White*, student ID: *A00234*, credits: *24*
* *Lisa Simpson*, student ID: *C22044*, credits: *56*
* *Charlie Brown*, student ID: *A12003*, credits: *6*

Enter all three into a lab and print a list to the screen.

A screenshot of a cell phone

Description automatically generated

Exercise 1.28: Use the inspector on a **LabClass** object to discover what fields it has. **See question 2.**

**A screenshot of a cell phone

Description automatically generated**

Exercise 1.29: Set the instructor, room, and time for a lab, and print the list to the terminal window to check that these new details appear.

**A screenshot of a cell phone

Description automatically generated**