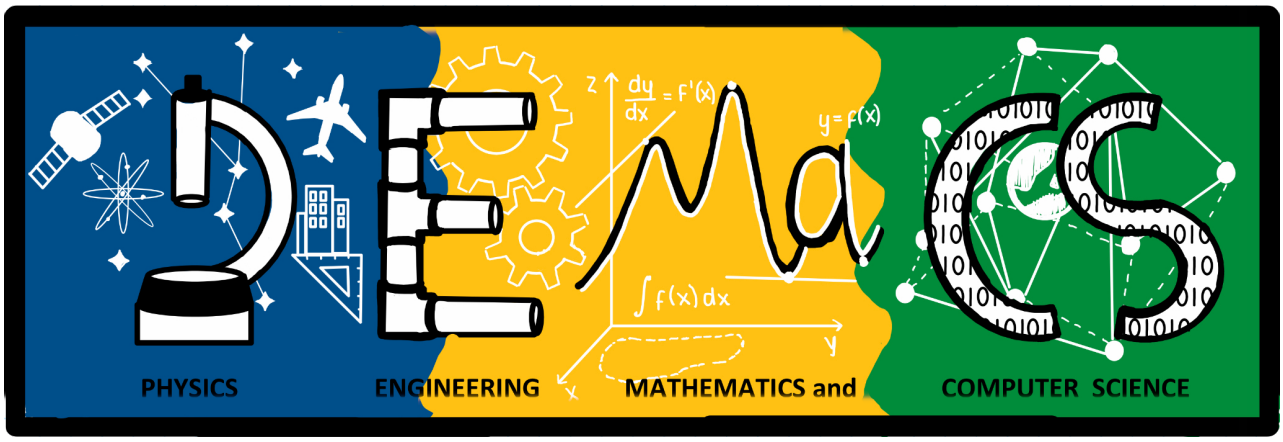


# Elements of Computer Programming II (CSCI-121) Java In-Class Activity: Introduction to Objects and Classes

## Learning Outcomes

By the end of this activity, a student should be able to:

1. Understand the difference between a class and an object.
2. Create objects from classes using BlueJ.
3. Interact with objects using the public methods.
4. Use parameters to pass information to methods.
5. Set the state of an object.

## Introduction

This activity will provide practice in problem-solving, working in a group, and algorithm design and implementation. It will also introduce the concepts of Object-Oriented Programming. Finally, the activity will introduce you to a simple integrated development environment (IDE) for Java that enables one to easily implement classes and instantiate objects.

## BlueJ IDE

In the previous lecture session, your instructor introduced the BlueJ IDE. This environment will be used to communicate the concepts of classes and objects. Eventually, the course will

transition to using more advanced and professional IDEs, in particular, IntelliJ.

## The Object-Oriented Paradigm

Up to now, all programs written in Python have focused on designing the actions or procedures and the logic first. The program is viewed as a logical set of functions that processes data and produces output. Object-oriented programming takes the object-first approach, where data are objects that have behavior; this behavior, when invoked, transforms the data to output. Logic and actions are incorporated into the behavior of the objects, and it is through object interaction that a computation can be realized.

## Activity

1. Consider the BlueJ project, Shapes, download as part of the Module 1 Supplements. Use this project to:
    - Instantiate various objects of varying sizes and colors. Pay particular attention to each object's default position and sizes once created. [2 minutes]
    - Discover the point of reference for each object used to locate its image on the screen. This will be important when trying to build an image made of a collection of objects. [2 minutes]
    - Experiment with the various methods available with each object to affect object behavior. Understand what each method does and how the parameters inform the behavior. [2 minutes]
2. You are required to instantiate the objects necessary to reproduce the following red image. The pixel dimensions are in black.

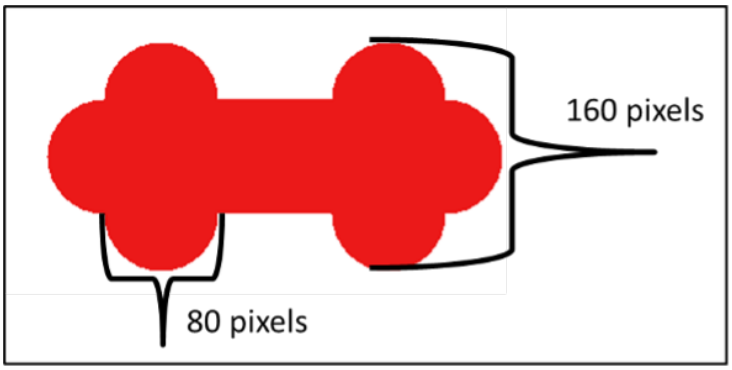


Figure 1: Image to be generated using the Shapes BlueJ project.

    - Working with your partner and the project documentation, design an algorithm to

generate the image shown in Figure 1. Use only the project documentation and your knowledge of the objects' default location and size. **Do NOT code first!**

- You may use the following pseudocode to write down the algorithm:
  - **ObjectName** *action* (value1, value2, ..., value*N*)
- Use the following pseudocode notation to signify the instantiation of an object of a particular type or class.
  - *Type* **ObjectName** = new *Type*()
- Give each object a name appropriate to its function.
- Implement your algorithm in the main method of a class called DogBone.

3. **On your own**: now that you have built familiarity with the classes and their objects, consider using the same approach as above to produce the following image of an isosceles trapezoid:

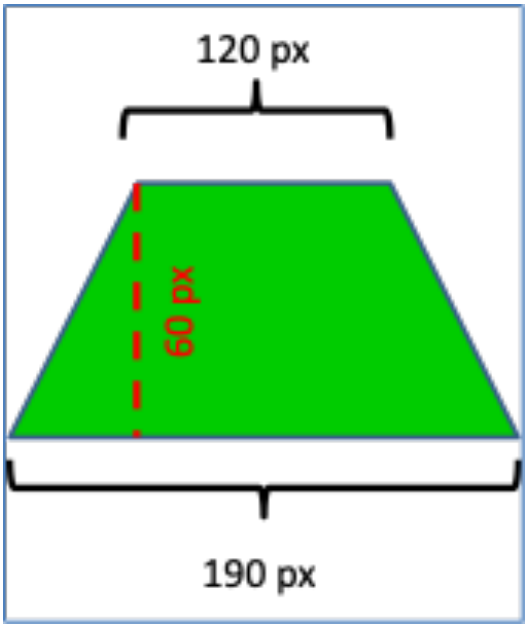


Figure 2: Trapezoid image to be generated using the Shapes BlueJ project.