Classes and Variables

- There’s an old saying that says, “Programs are algorithms plus data structures.” There’s a lot of truth to that but we can make it even more basic. Even the simplest programs boil down to two things:

1. Data
2. And actions on that data.

- Functions

- Subroutines

- Procedures

- Methods

- For example: Data of the string “Hello World!” and the action performed on that data: the printing of the string to the screen.

- The idea behind objects and object-orientation is how we organize the data and actions in our program.

- Each object represents some entity in the program. It could represent something from the domain of the program: student (at a school), monster (in a game), weapon (in a game), employee (in a payroll program), course.

- Objects could also represent a program element such as: a window, button, or data output stream.

- Whatever kind of thing the object represents, that object will hold the data related to that thing and the actions related to that data.

Example:

Employee

Name: “John Doe”

Department: “Information Technology”

Position: “Software Developer”

Pay rate: 25.00

Potential action: Calculate this employee’s pay.

A different employee would have the same actions and the same kinds of data, but different values.

Example 2:

Monster

Kind: “Goblin”

Hit points: 30

Base damage: 3-6 hit points

Location: 26, 45

Potential actions: Take damage, move, attack.

- This is a potential monster object for a game. Notice that it has different kinds of data and different potential actions.

- The actions of an object are not always the actions of the actual person or thing the object represents. If we look at example 1, we can see that employees don’t calculate their own pay. But an employee’s pay does depend on information specific to the employee. So, an employee object would do the calculation of pay because that calculation depends on data specific to that employee that is held in a specific employee object.

True or false: An object keeps track of the data about a specific item and allows operations on that data.

A. True

B. False

- Class: A template or pattern for a particular kind of object. The class will define the types of data a particular kind of object will contain and the variables names for those data items. A class also defines the actions (methods) that kind of object can perform.

- So to use objects, we’ll have classes to specify what the objects look like and what they can do. Then we’ll create actual objects and call their methods to get work done in our programs.

- Which of the following is the best analogy? Class is to object as

1. oven is to cookie.
2. cookie cutter is to cookie.
3. cookie is to oven.
4. cookie is to cookie cutter.