Task Five: Class Design

One of the most important aspects of object-oriented programming is correctly using classes in your programs.

A class is a description of an object. You can think of it as a blueprint. If you define a class, you can then create multiple objects of that class that follow the “blueprint”.

For example, if you made a class called “Fruit”, you could give the class certain properties and functions. Fruit could include a name and color. This means that every Fruit would have a name and a color. You could then make objects of type Fruit. All of these objects would have a name and color, but the differences in names and colors are what make the objects unique. For example, a Fruit could have name “Apple” and color “Red”, and another Fruit could have name “Pear” and color “Green”, but they would both be of type Fruit.

For this task, you will be designing your own set of classes. You will also be designing a driver program. For our purposes, the driver is where you actually create the objects and use their methods. It is the file that contains the main method (so it is the only file that is run).

On the next page, there is a walkthrough of the structure of a class file.

Task:

Your task is to create three classes: Backpack, Student, and School, and a driver program to test them.

* Backpack is a class with fields for its color, owner, and whether or not it is packed. It also has a method to pack it.
* Student is a class that has a Backpack, a name, a grade, and an age.
* School is a class that has five Students, a name, a type (elementary, middle, etc), and a founding year. School also has a method that returns the names of the students in the school and the name of the school.
* Your driver program must include the creation of three schools. Print out the names in the school and the students in each school.

Good luck, and remember to ask if you have any questions.

Example of a Class file:

public class Person

{

This is where you declare all of the private fields, or properties, of the class.

private int age;

private String name;

This is what is called a default constructor. It sets all of the values to what they will be as soon as the object is created.

public Person()

{

age = 0;

name = “”;

}

This is a constructor. It has two arguments in the method header. These two arguments are values that you can input when you make the object.

public Person(int a, String n)

{

age = a;

name = n;

}

This is an example of a modifier method. It allows you to change the value of a property once you have created an object.

public void setAge(int age1)

{

age = age1;

}

This is an example of a return method. It gives you the value of that specific field of an object.

public String getName()

{

return name;

}

}

Example of how to create an object of the class – in the driver program:

Person p = new Person(); This creates a person with default values

Person p2 = new Person(23, “Bob”); This creates a person whose age is 23 and whose name is “Bob”.