 

**Advanced Placement Computer Science**

[**Shenendehowa HS**](http://www.shenet.org/shen-high-school/)[**mr Hanley**](http://hanley.co.nr)

**Unit 4: Object Oriented Programming**

**Lesson: Classes**

***Last Updated:*** *10/24/2017*

Lesson: Parameter Passing Mechanisms

*Last Updated: 100/11001/1100*

Modern programs are often organized into separate \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_, each with its own responsibilities.

This began in the 1980’s with the desire to recreate more \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Classes act as software blueprints, allowing programmers to use some classes to create multiple objects, which contain their own \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ and \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

While there is only one class, one can create multiple \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

|  |  |
| --- | --- |
| Class | Object |
| One definition, contains logic that all objects share. Sometimes called behaviors. | **Can have multiple instances, each has its own unique data. Share all behaviors via the class** |
| Some classes are built on top of existing classes, this is called \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | **Executing programs use objects primarily, not classes. Objects send each other messages.** |
|  | **State of an object = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** |

How does a typical class manage information? There are 4 things inside classes

public class SomeClass {

 //Variable Section, typically private so only this class can modify

 //Constructors, special methods that \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ objects of this type

 //NOTE:

 public SomeClass() {

 }

 public SomeClass(String nm) {

 }

 //Accessors – must be public

 //Mutators – must be public

}
public class UserOfSomeClass {
 public static void main(String[ ] args) {
 SomeClass s1 = new SomeClass(); //instanciate an object called s1
 s1.setName(“Arthur”);
 s1.setAge(16);
 System.out.println(s1.getAge());
 //Use the overloaded constructor to create an object for s2 named “Sara”

 //Set her age to 17
 //Print out her information

}

Access specifiers

**private**

 **public

protected**What is a constructor?

What is a member variable?

What are overloaded methods?

For example, the following structure may apply to a java program;

FILE: Client.java
Contains: main method, which will use 3 Contacts called c1, c2 and c3

main() starts the program and has a main loop

FILE: Contact.java

Contains: no main method, has variables which pertain to a cell phone contact

1. public class Contact { //Represents a cell phone contact (Name and Phone Number)
2.
3.
4.
5.
6.
7.
8.
9.
10.
11.
12.
13.
14.
15.
16.
17.
18.
19.
20.
21.
22.
23.
24.
25.
26.
27.
28.
29.
30.
31.
32.
33.
34.
35.
36.
37.
38.
39.
40.
41.
42. public class Client {
43. public static void main(String[] args) {
44.
45.
46.
47.
48.
49.
50.
51.
52. }

54) }