

Key Points – Defining and Extending Classes

We have now learned how to define and extend classes. Here are the key points:

- A class definition includes the declaration of variables, known as *fields*, and *methods*.
- The methods and fields in a class are the *members* of the class.
- A object may be printed by passing it as an argument to the `println()` method. It is automatically converted to a string using a method called `toString()`.
- All Java classes are subclasses of the `Object` class.
- The `Object` class has a `toString()` method; however, a class may override this by providing its own `toString()` method.
- A class may include more than one method with the same name, provided the type of the return value or the number and/or type of the arguments are unique. This feature is known as *method overloading*.
- A `public` class member (field or method) is accessible anywhere the class is accessible.
- A `private` class member is accessible only within the class where it is defined.
- A `protected` class member is accessible within its class and within any class that extends that class.
- A class is defined as a subclass of another using an `extends` clause.
- A class that extends another class inherits characteristics (fields and methods) from the class it extends. This feature is known as *inheritance*.
- Classes provide an inherent ability to protect and hide data fields from other classes that might use the class. This feature is known as *encapsulation*.
- A class that extends another class typically uses the `super()` method — the superclass constructor — to construct the portion of it defined in its superclass.
- In a class definition, the reserved word `this` is a reference to the current (receiving) object.
- An object of an extended class can be used anywhere an object of its superclass can be used. This feature is called *polymorphism*.
- An instance method is called through an instance of its class.
- A static method is *not* called through an instance of a class.
- A "static method" is also called a "class method".
- Java "programs" are classes that can be instantiated as objects (although this is rarely done).
- Java "classes" (from which objects can be instantiated) can also be executed, provided they include a `main()` method.
- All Java programs have a class that "drives" the application. This class must have a method called `main()` which serves as the entry point for the program.

References

- [1] K. Arnold, J. Gosling, *The Java programming language* (Reading, MA: Addison-Wesley, 2nd ed., 1997).