Every class you declare represents a new type in Java.

The public methods of a class are also known as the class's public services or public interface. The primary purpose of public methods is to present to the class's clients a view of the services the class provides. Clients of the class need not be concerned with how the class accomplishes its tasks. For this reason, private class members are not directly accessible to the class's clients.

An object that contains consistent data (consistent state) has data values that are always kept in range.

A value passed to a method to modify an instance variable is a correct value if that value is in the instance variable's allowed range. A correct value is always a consistent value, but a consistent value is not correct if a method receives an out-of-range value and sets it to a consistent value to maintain the object in a consistent state.

String class static method format is similar to method System.out.printf except that format returns a formatted String rather than displaying it in a command window.

All objects in Java have a toString method that returns a String representation of the object. Method toString is called implicitly when an object appears in code where a String is needed.

A non-static method of an object implicitly uses keyword this to refer to the object's instance variables and other methods. Keyword this can also be used explicitly.

The compiler produces a separate file with the .class extension for every compiled class.

If a method contains a local variable with the same name as one of its class's fields, the local variable shadows the field in the method's scope. The method can use the this reference to refer to the shadowed field explicitly.

Overloaded constructors enable objects of a class to be initialized in different ways. The compiler differentiates overloaded constructors by their signatures (parameter lists).

Every class must have at least one constructor. If none are provided, the compiler creates a default constructor that initializes the instance variables to the initial values specified in their declarations or to their default values.

If a class declares constructors, the compiler will not create a default constructor. To specify the default initialization for objects of a class with multiple constructors, the programmer must declare a no-argument constructor.

Set methods are commonly called mutator methods because they typically change a value. Get methods are commonly called accessor methods or query methods. A predicate method tests whether a condition is true or false.

A class can have references to objects of other classes as members. Such a capability is called composition and is sometimes referred to as a has-a relationship.

All enum types are reference types. An enum type is declared with an enum declaration, which is a comma-separated list of enum constants.

```java
enum GameStatus { WON, LOST, CONTINUE }
```

enum types are implicitly final, because they declare constants that should not be modified.

enum constants are implicitly static.

Any attempt to create an object of an enum type with operator new results in a compilation error.

enum constants can be used anywhere constants can be used, such as in the case labels of switch statements and to control enhanced for statements.

Every class in Java has the methods of class Object, one of which is the finalize method.

The Java Virtual Machine (JVM) performs automatic garbage collection to reclaim the memory occupied by objects that are no longer in use. When there are no more references to an object, the object is marked for garbage collection by the JVM. The memory for such an object can be reclaimed when the JVM executes its garbage collector.

The finalize method is called by the garbage collector just before it reclaims the object's memory. Method finalize does not take parameters and has return type void.
• The garbage collector may never execute before a program terminates. Thus, it is unclear if, or when, method finalize will be called.

• A static variable represents classwide information that is shared among all objects of the class.

• Static variables have class scope. A class's public static members can be accessed through a reference to any object of the class, or they can be accessed by qualifying the member name with the class name and a dot (.). A class's private static class members can be accessed only through methods of the class.

• Static class members exist even when no objects of the class exist—they are available as soon as the class is loaded into memory at execution time. To access a private static member when no objects of the class exist, a public static method must be provided.

• System class static method gc indicates that the garbage collector should make a best-effort attempt to reclaim objects that are eligible for garbage collection.

    System.gc()  //explicitly call the Java garbage collector

• A method declared static cannot access non-static class members, because a static method can be called even when no objects of the class have been instantiated.

• The this reference cannot be used in a static method.

• A static import declaration enables programmers to refer to imported static members without the class name and a dot (.). A single static import declaration imports one static member, and a static import on demand imports all static members of a class.

• In the context of an application, the principle of least privilege states that code should be granted only the amount of privilege and access that the code needs to accomplish its designated task.

• Keyword final specifies that a variable is not modifiable—in other words, it is constant. Constants can be initialized when they are declared or by each of a class’s constructors. If a final variable is not initialized, a compilation error occurs.

• Software is constructed from existing, well-defined, carefully tested, well-documented, portable, widely available components. Software reusability speeds the development of powerful, high-quality software. Rapid application development (RAD) is of great interest today.

• Java programmers now have thousands of classes in the Java API from which to choose to help them implement Java programs. The Java API classes enable Java programmers to bring new applications to market faster by using preexisting, tested components.

• The client of a class cares about the functionality the class offers, but not about how the functionality is implemented. This is referred to as data abstraction. Although programmers may know the details of a class’s implementation, they should not write code that depends on these details. This enables a class to be replaced with another version without affecting the rest of the system.

• An abstract data type (ADT) consists of a data representation and the operations that can be performed on that data.

• Each class in the Java API belongs to a package that contains a group of related classes. Packages help manage the complexity of application components and facilitate software reuse.

• Packages provide a convention for unique class names that helps prevent class name conflicts.

• Before a class can be imported into multiple applications, the class must be placed in a package. There can be only one package declaration in each Java source-code file, and it must precede all other declarations and statements in the file.

• Every package name should start with your Internet domain name in reverse order. After the domain name is reversed, you can choose any other names you want for your package.

• When compiling a class in a package, the javac command-line option -d specifies where to store the package and causes the compiler to create the package's directories if they do not exist.

• The package name is part of the fully qualified class name. This helps prevent name conflicts.

• The compiler uses a class loader to locate the classes it needs in the classpath. The classpath consists of a list of directories or archive files, each separated by a directory separator.

• If no access modifier is specified for a method or variable when it is declared in a class, the method or variable is considered to have package access.