

Student Name _____

- There are two kinds of types in Java: primitive types and reference types. The primitive types are: `byte`, `short`, `int`, and `long` for integers; `float` and `double` for real values; `Boolean` for logical values; and `char` for individual characters. Reference types are (usually) class types.
- The type of an object must be a reference type.
- An identifier may begin with a letter or `_` (underscore), which may be followed by a number of these characters or digits; it may not be a Java keyword.
- Use meaningful identifiers that suggest what they represent, because such names make a program easier to read and understand.
- Java is case sensitive.
- Any name in a program that is not a Java keyword is an identifier and must be declared before it can be used.
- The value of a primitive - type variable is a value of that type; the value of a reference - type variable is an address of a memory location where a value of that type can be stored.
- If an initial value is not specified in a variable declaration, a default value will be used. This is zero for numeric types, `false` for `Boolean`, and `null` for reference types.
- The `final` modifier is used to declare that a value stored in a memory location cannot be changed during program execution.
- Using named constants instead of the literals they represent improves code readability and facilitates program maintenance.
- Placing constant declarations at the beginning of the class or method in which they are used is good programming practice because it makes it easy to locate them when modifications are necessary.
- Every program should contain opening, documentation that describes what the program does, its input and output, special techniques and algorithms it uses, special instructions for using program, the programmer's name, the date the program was written, and a history of its modification. A multi - line comment enclosed between `/*` and `*/` can be used for this.
- Inline comments begin with `//` and run to the end of the line. They are useful to list imported classes, to add information about a variable or constant declaration, to identify key program statements, and to explain difficult or obscure sections of code.

Student Name _____

- A class is a pattern or blueprint that is used to model real - world objects that have similar attributes. The form of a class is:

```
class className extends existingClassName
{
    // Variables (and constants): determine an object's
    // attributes / state

    // Methods: determine an object's behavior
}
```
- Every Java application must have a main method since this is where program execution begins.
- Import statements allow one to refer to classes in a package by their simple names - the single identifiers used in the package - instead of using their fully - qualified names of the form `package_name.ClassName`.
- The classes in `java.lang` are imported automatically; no import statement is needed.
- The Java Application Programming Interface (API) is a collection of more than 1,600 classes organized into packages. Hypertext - based documentation accessible via a Web browser is an extremely valuable tool for finding information about these classes.
- Java's `Object` class is at the root (top) of the API class hierarchy and is thus a superclass of every class.
- A class that extends another class inherits the attributes (fields) and behaviors (methods) of that class. Thus, in searching the API documentation for information about classes, one should also look at what they inherit from superclasses.
- In designing GUI applications, it is helpful to draw transition diagrams to describe the program's behavior - different windows that can appear and the user actions that cause changes from one to another.
- The `exit()` method in Java's `System` class can be used to terminate execution of a program.
- Input - and output - dialogs provided by Java's Swing `JOptionPane` class provide a fairly simple, yet powerful, way to get information from and to the user in a GUI environment.