Java Objects: Declare, Init, Use

Declaration

Java is a strongly-typed language, meaning all variables need to be declared as a particular type. This includes:

local variables, parameters, data members, even return value of function.

Programming languages have **built-in types and programmer-defined types**

built-in types are scalars. In Java: int, float, double, char, boolean, string

When you write a class, you create a programmer-defined type.

With a declaration, the programmer specifies the existence and type of a variable

```
int x; // int is the type, x is the variable name

Person person; // Person is the type, person the variable(instance)
```

Shorthand: You can declare and initialize on the same line:

```
int x = 3;
Person person = new Person();
```

Initialization

You can initialize a variable after the call to new, e.g.

```
Person p = new Person();
p.age=30;
p.name="Bob";
```

Sometimes it’s more elegant to use a constructor with parameters.

A constructor is a special method in Java. It is special because:

- It has the same name as the class.
- It has no type associated with it.
- It is called in a special way: when an object is created with new.
Often, a class will have multiple constructors:
   One with no parameters that sets default values for data members
   Others with parameters that allows the data members to be set on new.

**Using an object once it’s initialized.**
Once you have an object, you can:

1. set/get its data members
   
   person.age=56;

2. call methods on it.
   
   old= person.older(otherPerson)

3. print it
   
   System.out.println(person)

Note that the Java equivalent to Python’s `__str__` is `toString`:

```java
public String toString()
{
    // return a string that should be printed out for the object
    // often this is a concatenation of data members
    // if we were in class Person…
    return this.name+"":"+this.age
}
```

**Instructor code sample**

Write a class Car that represents a car in a race.

Declare data members: startPosition, currentPosition

Write a sample main creating a car and setting data members

Write a constructor to make it easier.

Write a `toString()` method