Overloading (Methods)

Same name, different parameter(s) or return type

Order of operations

1. Exact match- same return type, argument count, and argument type
2. Conversion- primitive to primitive, wrapper to wrapper
3. Boxing/Unboxing- from primitive data type to wrapper, or vice versa
4. Variable Length Arguments (Varargs)- same return type and argument type, different argument count

Variable Scopes ( {} defines scope)

Global in Java? NO

Everything must belong to a class or object

Static or Class scope

Accessible to all members of that class

Keyword “static”

Lifetime of the class

Has default values

Method or Local scope

Accessible within the method declared in

Do NOT get default values

Block Scope

Accessible within the block it was created or in a control statement (if/for/do)

Do NOT get default values

Instance Scope

Accessible inside a particular object

Has default values

Lifetime of the object it belongs to

Shadowing – When variables in different scopes have the same name

public String setName(String name){

this.name = name;

} Instance Scope Method Scope

Final

Keyword “final”

Can have variables, methods, and classes as final

Variable- Cannot be changed

Method- Cannot be overridden

Class- Cannot be extended