

Objects and classes

I. Review

- a) _____ typing: operations will only be executed on values of compatible types
- b) _____ typing: type compatibility is checked at compile-time
 - i) Variables must be declared with their type
- c) Primitive types
- d) Casting

II. History

- a) Improvements in tools, materials, processes allow larger, more complex structures to be built in less time
- b) Difficult to build large, reliable programs using early programming languages
- c) Simula: first object-oriented language; used for simulation
- d) Smalltalk: Popularized object-oriented programming

III. Object-oriented programming

- a) Model objects from real world
- b) Helps identify features, even in intangible settings
- c) Easier to adapt programs to changes
- d) Easier to reuse code in other contexts

IV. Objects and classes

- a) Characterized by state and _____
- b) _____: template for making new objects
- c) Object: instance of class
- d) Visualize as a folder

v. Syntax

- a) _____: Code between curly braces (including class definition)
- b) _____: Variable that every instance will have a unique version of
- c) _____: Function or procedure providing behavior
 - i) Invoke with period after variable or expression referencing target object

vi. Variables

- a) Store pointer to object (address of object's folder)

vii. Fields

- a) Store an object's state
- b) Have a default value if not explicitly initialized

viii. Methods

- a) Provide an object's behavior
- b) Procedure: method with no return value
 - i) Return type: _____
- c) Return statement

IX. Scope

- a) Governs lifetime and accessibility of variable
- b) Inside-out rule

X. Overloading

- a) Methods with the same name, but different parameter types
- b) Signature: name, parameter types; must be distinct
 - i) Method overloads can | cannot be distinguished by return type
- c) Which version to call is determined at _____-time

XI. Reading topics (JHT)

- a) Class definition, object
- b) Reference, pointer
- c) Field
- d) Method, parameter, argument, method call
- e) Return statement
- f) Scope, inside-out rule
- g) new expression, instantiate
- h) Overload, signature