OOP & OOPLs

Rushikesh K Joshi
IIT Bombay

OOP in a nutshell

- The Object Abstraction
- Interfaces and Implementation
- Encapsulation and Visibility Control
- Inheritance and Polymorphism
- Generalization and Specialization
- Dynamic Binding

Where do OOPLs Differ?

- Objects & Classes
- Encapsulation
- Inheritance Models
- Firstclass features
- Purity of object orientations
- Typing issues
- Exception handling Mechanism
- Parameter passing .....

Where do OOPLs Differ?

- Threading
- Portability
- Packaging
- System Interface
- Trees vs forest

Classes & Objects

- Are there classes?
- Are classes objects?
- Do classes have classes?
- Nesting of classes?

Classes of Classes?

- Are there classes of classes?
- If not, how does the language handle the missing features?

Encapsulation

- Level of encapsulation
- Visibility model and control
- Can encapsulation be broken?

Inheritance Models

- Single Vs. Multiple
- Shared Vs. Repeated
- Dynamic binding or not?

First class features

- What all can be created, passed as input parameter and returned?

Purity of Object Orientation

- Functions vs. member functions
- Types
- Main
- Breakage of encapsulation
- Control constructs
- Exception handling
- System Interface ...

Typing Issues

- primitive vs. object types
- subtyping rules for member functions
- narrowing and widening
- Object type
- variables are typed or not: static vs dynamic typing
- Type safety

Exception handling

- Built in?
- Must or optional?
- Object oriented?

Parameter passing

- by reference, by value?
- keyword parameters?
- accessor specifiers ..

Threading

- built in?
- Models of threading?

Portability &Networkability

- Byte codes and interpretation
- Standardization
- Networkability
- Security

Packaging

- Packaging features?

- Files?

Trees vs. forest

- class Object?

Other features

- basic types
- libraries
- development environmen
- contracts and assertions
- reflection
- genericity ..