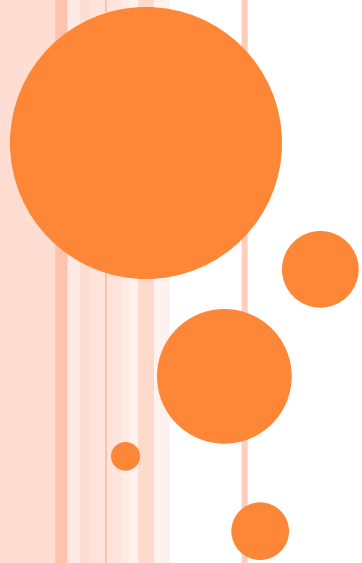


SOFTWARE ENGINEERING



LECTURE-41

Object Oriented Testing



TOPICS COVERED

- TRADITIONAL VS OO SW DEVELOPMENT AND TESTING
- OO CONCEPTS/EFFECT ON TESTING
- PROPOSED OO INTEGRATION TESTING APPROACH
- EXAMPLE USING TEST APPROACH
- CONCLUSION



OBJECT ORIENTED TESTING

SYSTEM TESTING

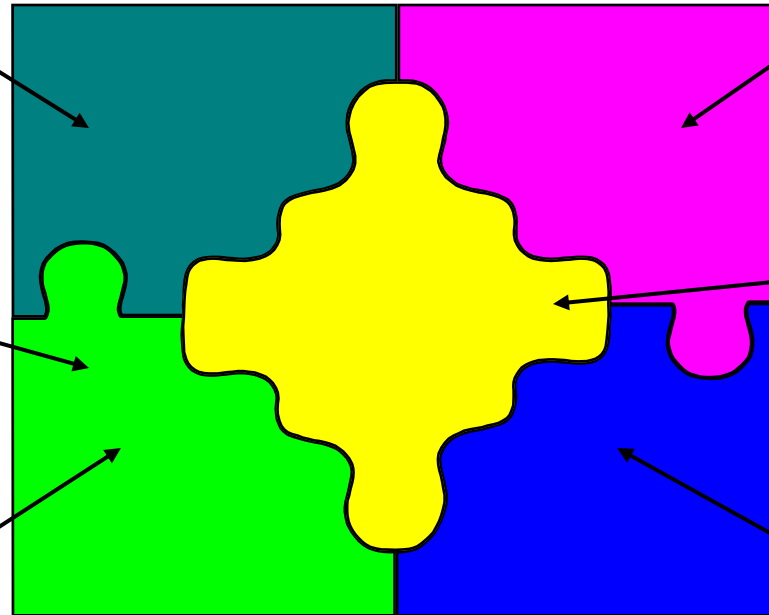
INHERITANCE

INTEGRATION TESTING

POLYMORPHISM

UNIT TESTING

ENCAPSULATION



By
Maj Nicko Petchiny



REFERENCES

- Developing an OO Software Testing and Maintenance Environment (King, Gao, Hsia, et-al)
- Incremental Testing of OO Class Structures (Harrold, McGregor)
- OO Integration Testing (Jorgensen, Ericksen)
- OO Software Testing, A Hierarchical Approach (Siegel)



TRADITIONAL DEVELOPMENT & TESTING (WATERFALL LIFE CYCLE)

- REQUIREMENTS SPEC TESTING SYSTEM
▶
- PRELIMINARY DESIGN INTEGRATION TESTING
 - FUNCTIONAL
 - DECOMPOSITION
- DETAILED DESIGN UNIT TESTING
▶



TRADITIONAL TESTING

- SYSTEM

- VERIFY SW SATISFIES ALL SW REQRS

- INTEGRATION

- BASED ON STRUCTURE OF DESIGN
- TOP DOWN OR BOTTOM UP APPROACH

- UNIT

- ENCAPSULATES FUNCTIONALITY



OO DEVELOPMENT & TESTING

- DEVELOPMENT BASED ON BEHAVIOUR
- COMPOSITION
- TYPICALLY RAPID PROTOTYPING
- INCREMENTAL APPROACH
- 3 TRADITIONAL TESTING LEVELS ARE NOT AS CLEARLY DEFINED



OBJECT ORIENTED TESTING

○ SYSTEM

- SAME AS TRADITIONAL
- STILL BASED ON REQRS SPEC

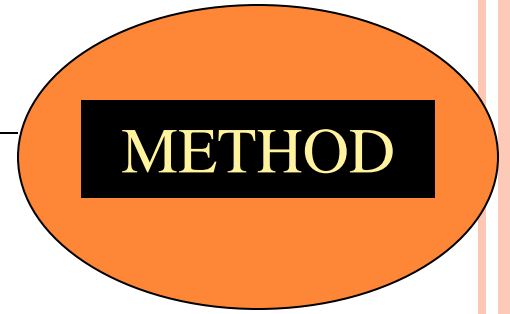
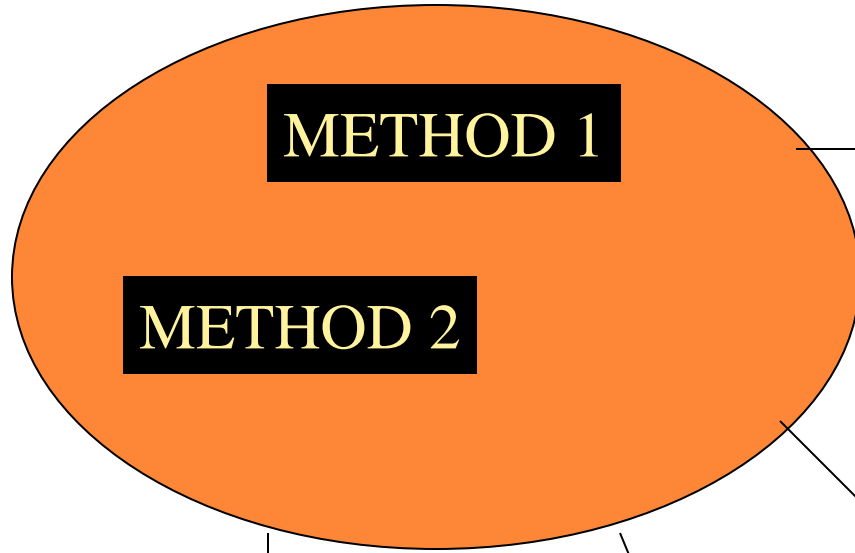
○ UNIT

- TWO COMMON STRUCTURES USED
 - METHOD*
 - CLASS
- SAME AS TRADITIONAL(DRIVERS & STUBS)

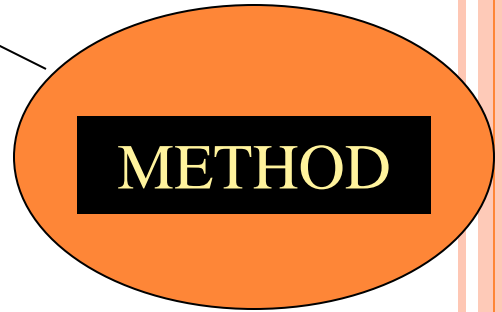


OBJECT CLASS A

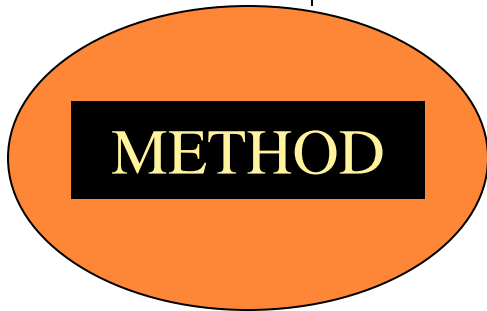
F



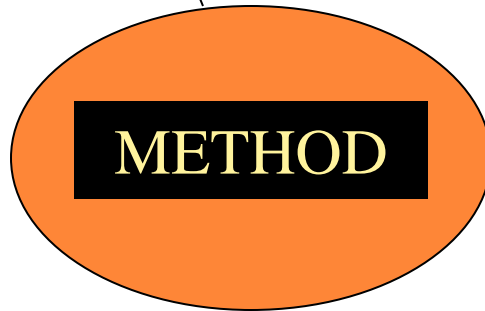
E



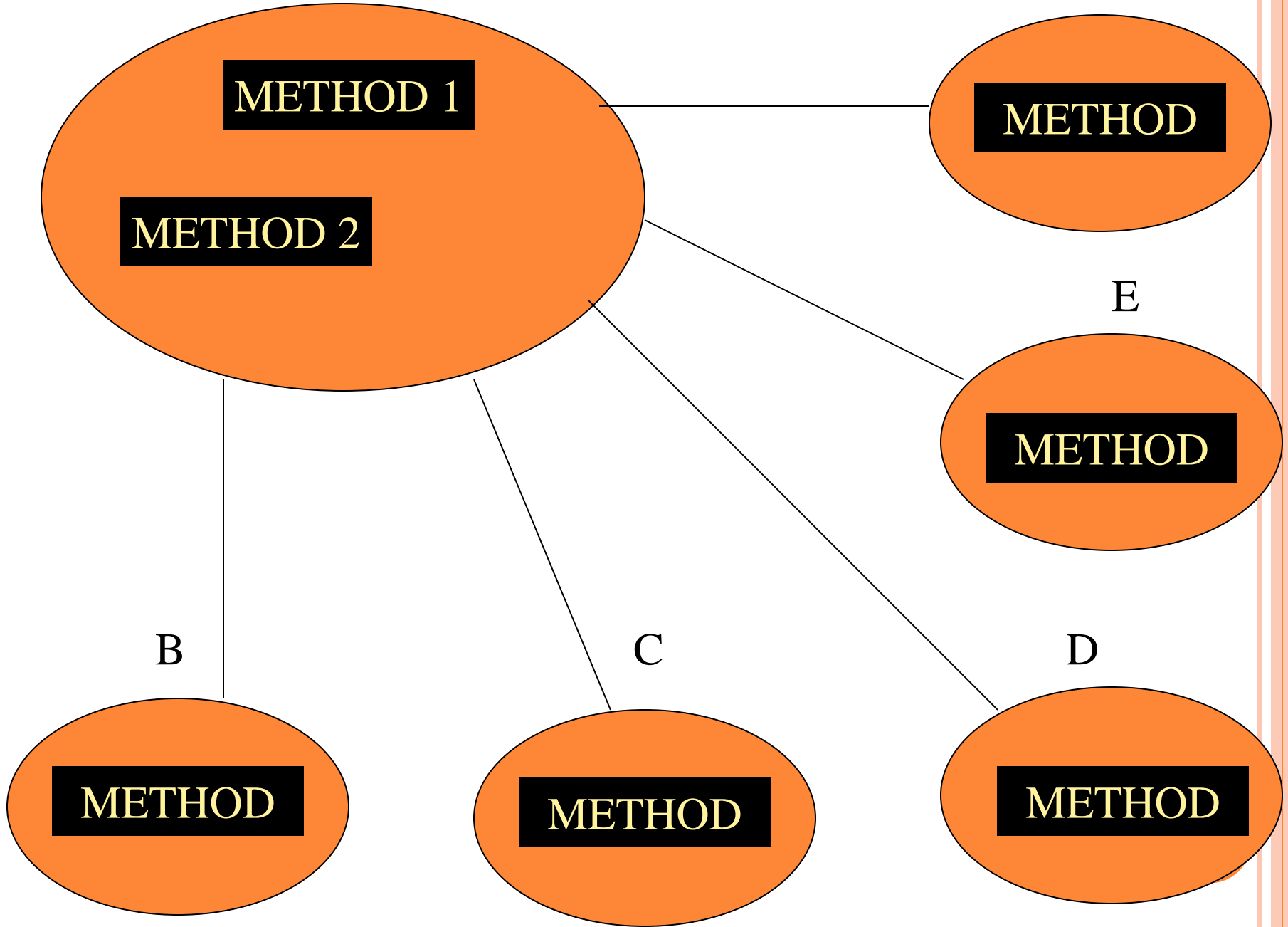
B



C

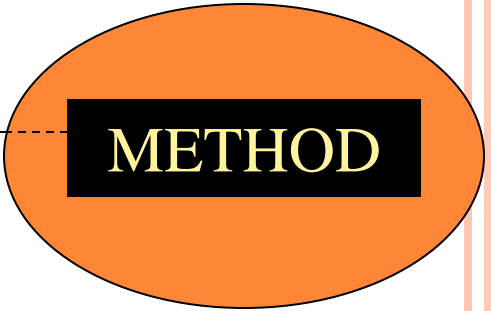
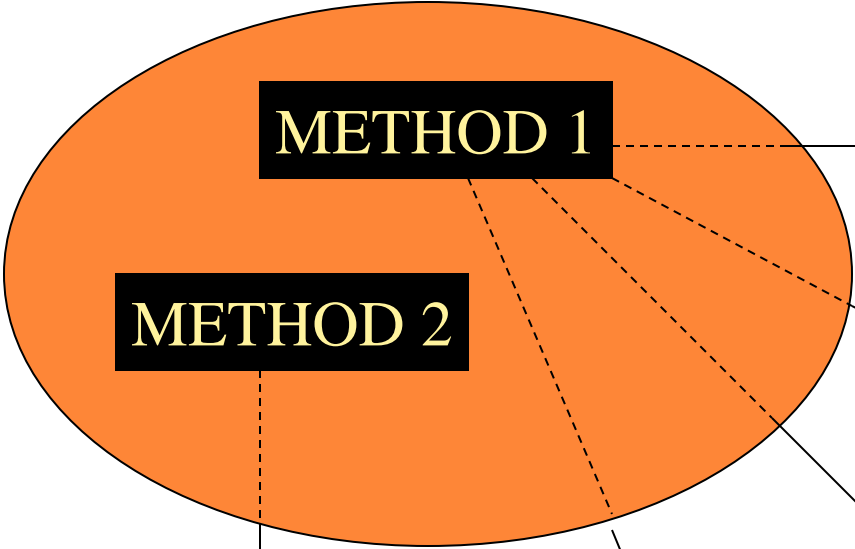


D

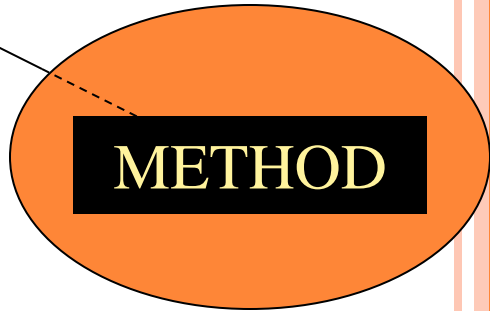


OBJECT CLASS A

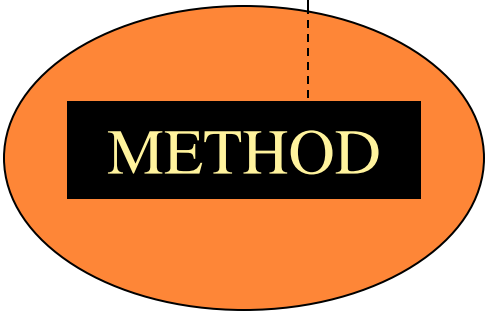
F



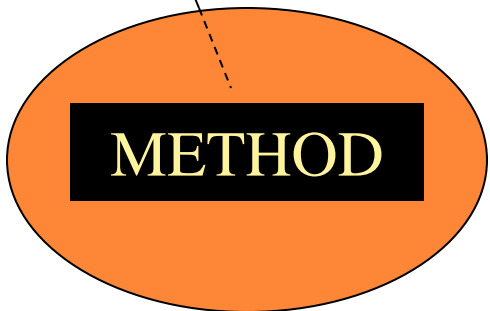
E



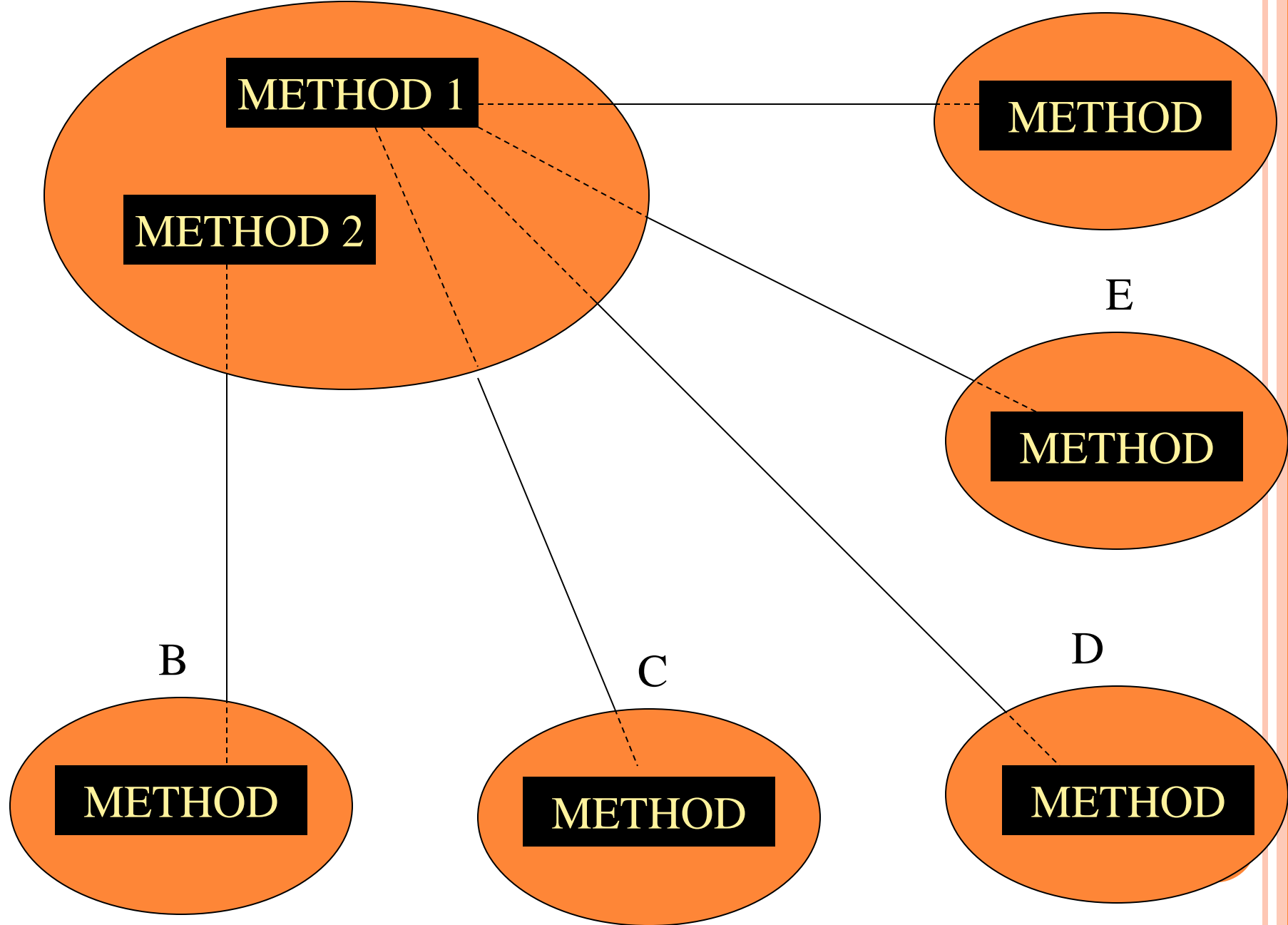
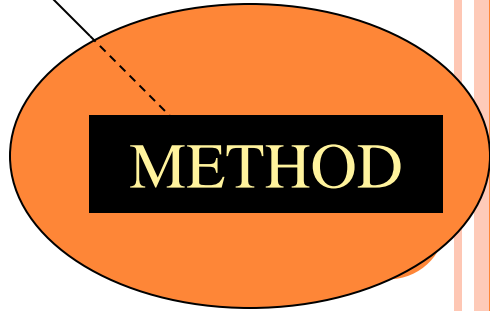
B



C



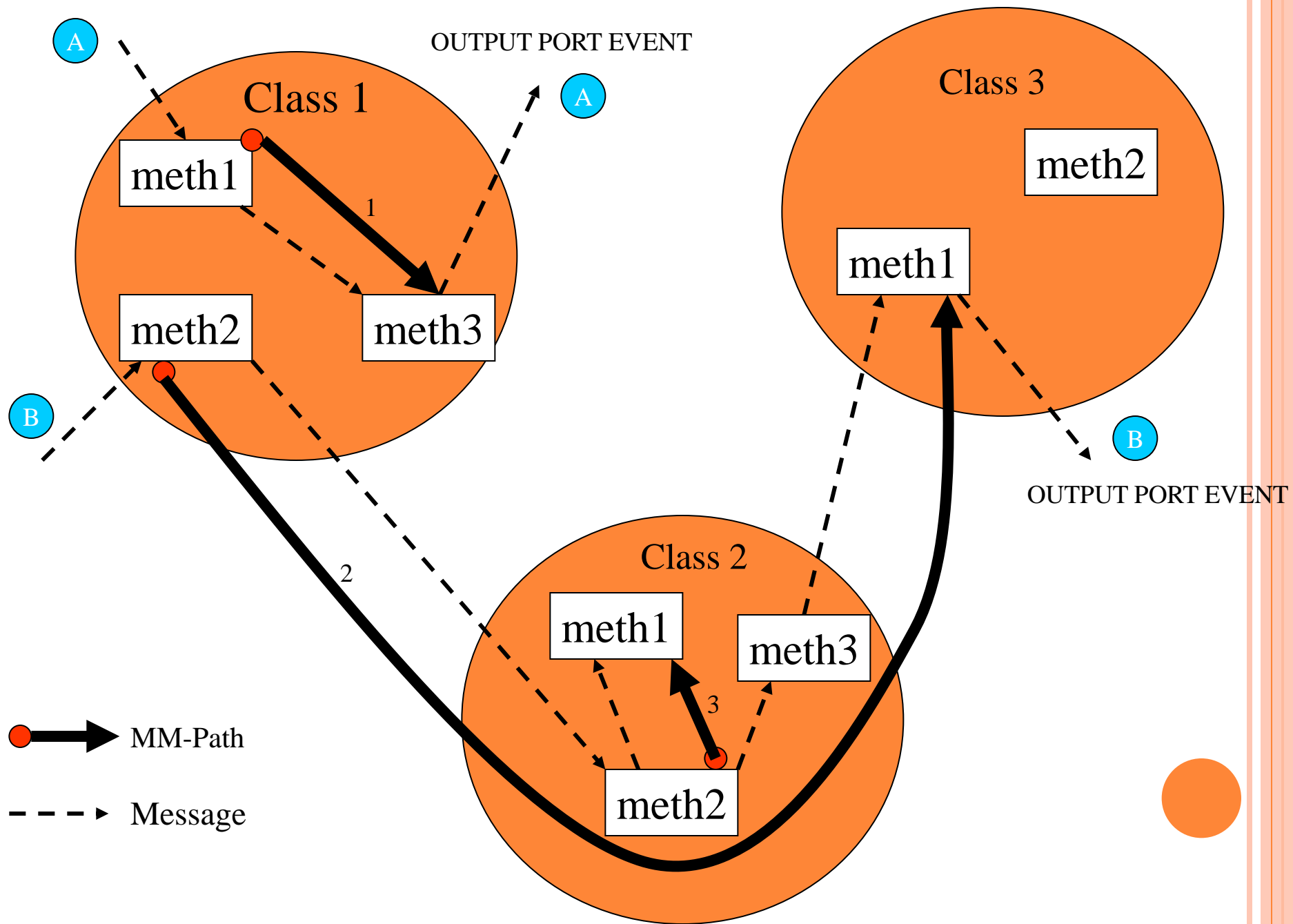
D



OO INTEGRATION TESTING

- MAIN PROGRAM IS MINIMIZED
- MOST COMPLICATED PART OF OO TESTING
- TESTING BASED ON COMPOSITION IN BOTTOM UP APPROACH
- USE OF CLUSTERS
- ORD - CLASS DEPENDENCIES
- BBD OR DIRECTED GRAPHS - SHOWS METHOD DEPENDENCIES





OO CONCEPTS/EFFECTS ON TESTING

- ENCAPSULATION
- POLYMORPHISM
- INHERITANCE



ENCAPSULATION

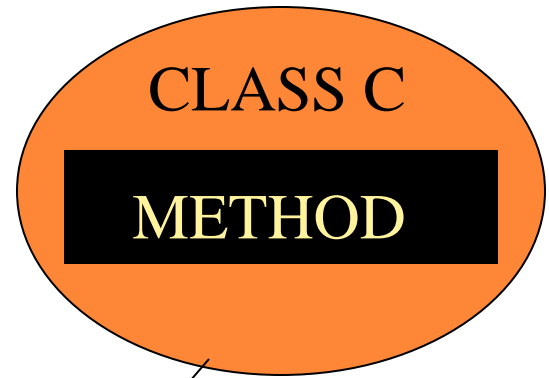
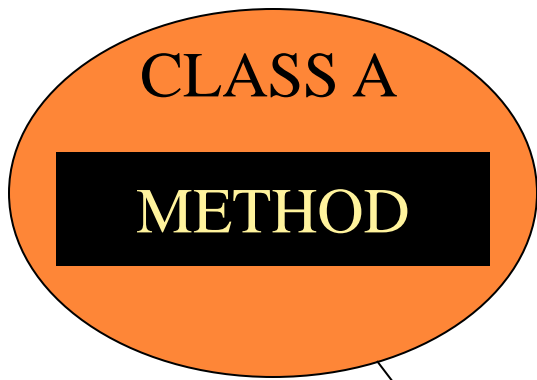
- CLASS STRUCTURE
- INTERFACE DEFINED BY PUBLIC METHODS
- BEHAVIOR DEFINED BY METHODS THAT OPERATE ON ITS INSTANCE DATA (IN CONVENTIONAL SEPARATE)
- HELPS ENFORCE INFO HIDING



ENCAPSULATION TESTING ISSUES

- MINIMIZES RIPPLE EFFECT (AT THE UNIT LEVEL) OF MAKING A CHANGE
- HIGHLY DELOCALIZED
 - CHANGE COULD RESULT IN SIGNIFICANT REGRESSION TESTING
- ORDER OF TESTING IS IMPORTANT (CAN REDUCE TESTING EFFORT)



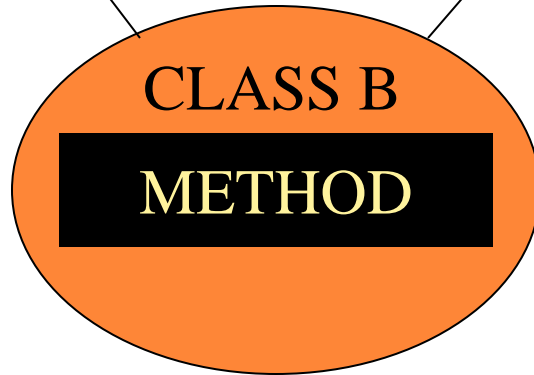


USES

An arrow pointing from the bottom of CLASS B to the bottom of CLASS A.

USES

An arrow pointing from the bottom of CLASS B to the bottom of CLASS C.



POLYMORPHISM

- AN ATTRIBUTE MAY HAVE MORE THAN ONE SET OF VALUES
- AN OPERATION MAY BE IMPLEMENTED BY MORE THAN ONE METHOD (e.g GRAPHICS)
- OVERLOADING (type or number of variables)
- DYNAMIC BINDING



OO TESTING ISSUES

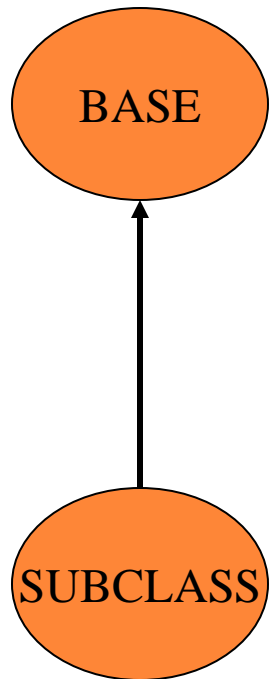
○ POLYMORPHISM

- DO YOU TEST ONE VARIANT ?
- DO YOU TEST ALL VARIATIONS ?
- IF ALL, DO YOU TEST ALL VARIANTS AT ALL LEVELS
 - UNIT
 - “INTEGRATION” OR SYSTEM LEVEL
- REUSE DRIVERS AND STUBS

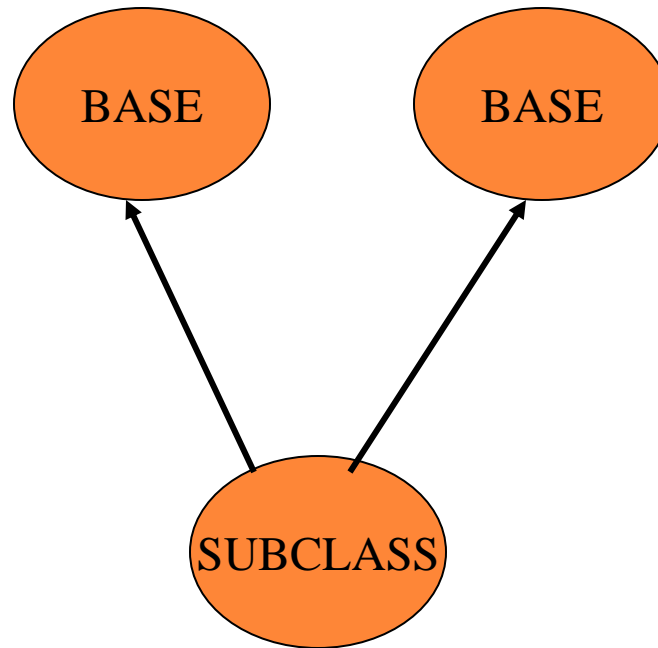


INHERITANCE STRUCTURES

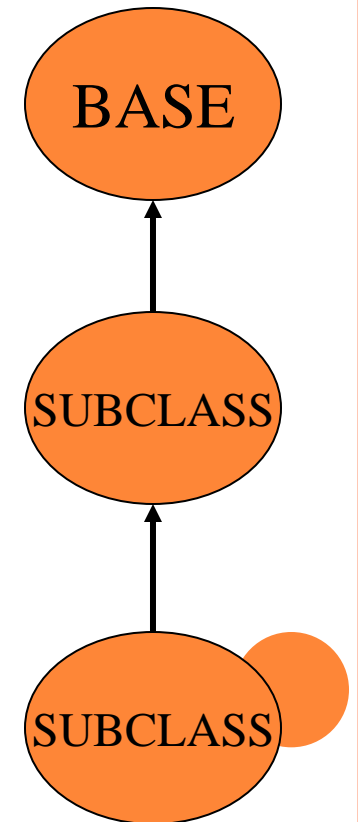
SINGLE



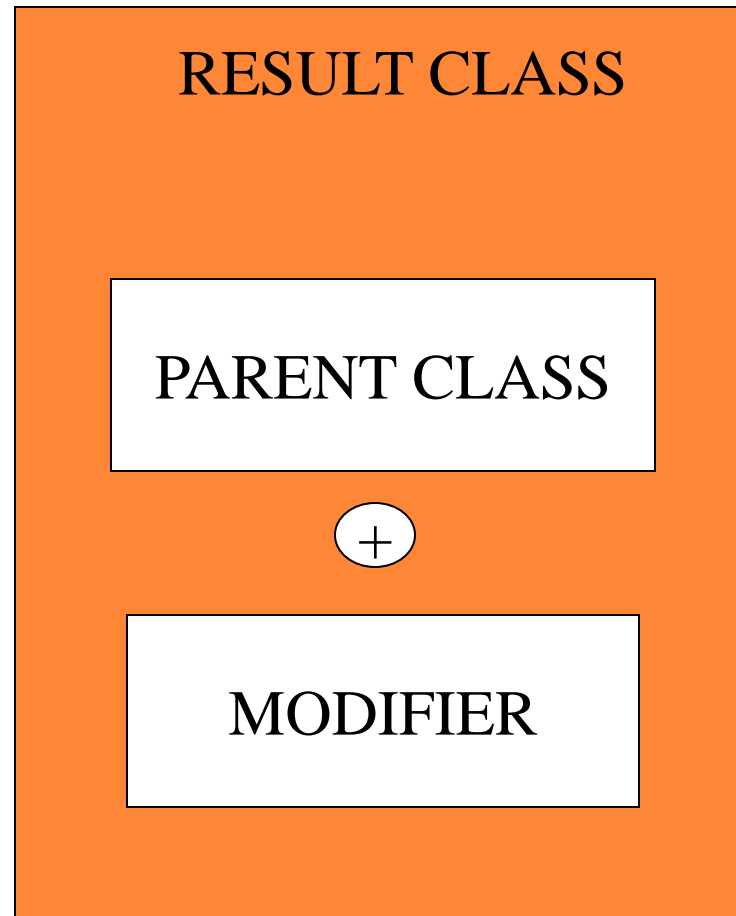
MULTIPLE



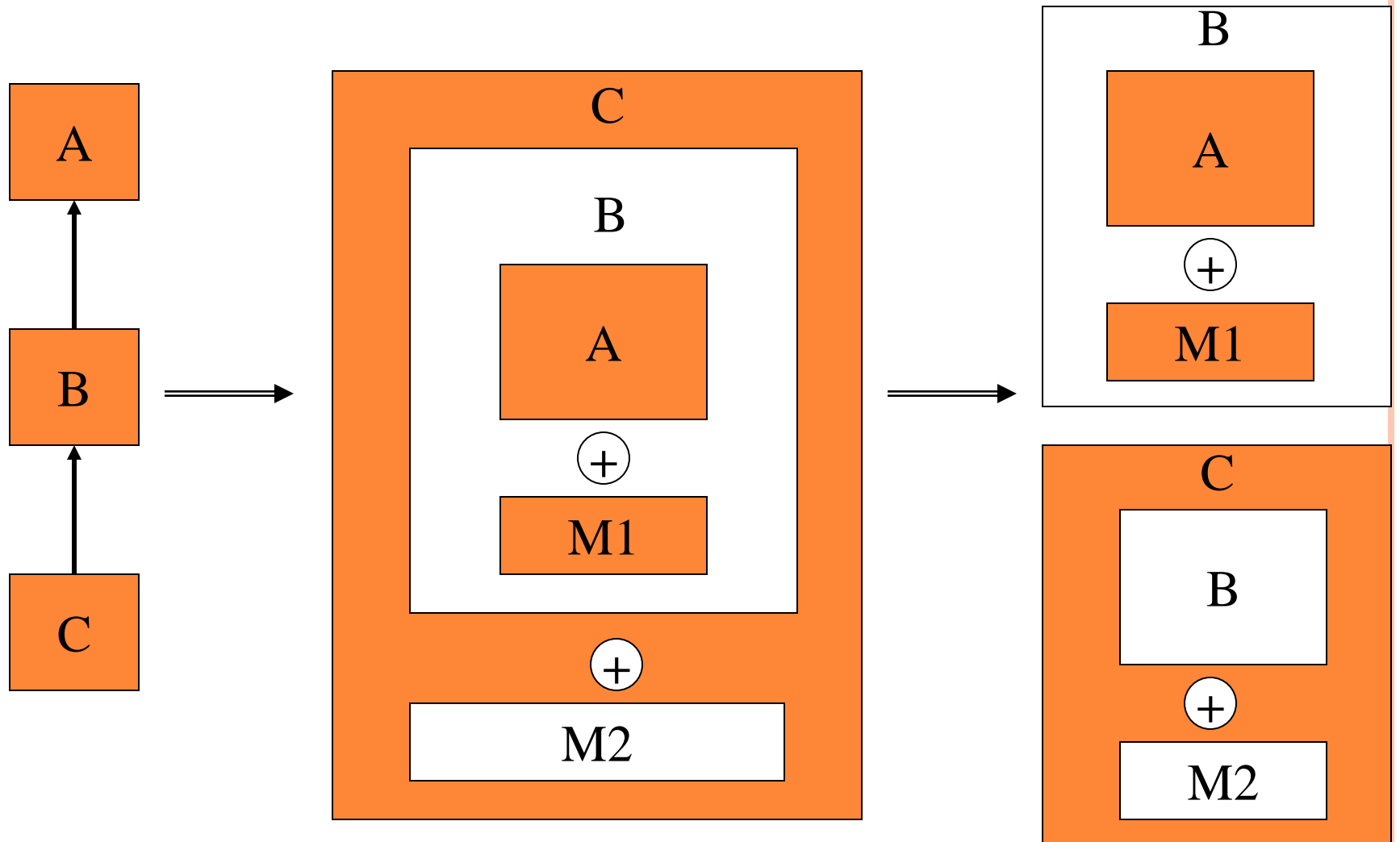
MULTIPLE LEVELS



INHERITANCE



INHERITANCE



INHERITANCE MODIFIERS

- NONE (ONLY INHERITED ATTRIBUTE)
- ADD NEW ATTRIBUTE(S)
- REDEFINE PARENT'S ATTRIBUTE(S)
- VIRTUAL ATTRIBUTE (THREADS IN JAVA)



OO TESTING ISSUES

○ INHERITANCE

- DO YOU COMPLETELY TEST ALL BASE CLASSES AND THEIR SUB-CLASSES ?
- DO YOU COMPLETELY TEST ALL BASE CLASSES AND ONLY TEST THE CHANGES OR MODIFICATIONS IN THEIR SUB-CLASSES ?
- AT WHAT LEVELS DO YOU TEST?
- IN WHICH ORDER DO YOU TEST?



INHERITED TESTING

SCENARIO	UNIT	INTEGRATION
NONE		X?
NEW	X	X?
REDEFINED	X	X
VIRTUAL (COMPLETED BY SUBCLASS)	X	X?
VIRTUAL (NOT COMPLETED)		



OO TESTING METHODOLOGY

- JORGENSEN AND ERICKSEN PROPOSE 5 LEVELS
- A METHOD - UNIT TESTING
- MESSAGE QUIESCENCE - INTEGRATION
- EVENT QUIESCENCE - INTEGRATION
- THREAD TESTING -SYSTEM
- THREAD INTERACTION -SYSTEM



CONSTRUCT DEFINITIONS

- MM-PATH (METHOD MESSAGE - PATH)
[MESSAGE QUIESCENCE]
 - SEQUENCE OF EXECUTIONS LINKED BY MESSAGES.
 - STARTS WITH METHOD AND ENDS WITH A METHOD THAT DOESN'T PRODUCE A MESSAGE



CONSTRUCT DEFINITIONS

- ASF (ATOMIC SYSTEM FUNCTION) [EVENT QUIESCENCE]
 - REPRESENTS AN INPUT EVENT
 - FOLLOWED BY A SET OF MM-PATHS
 - TERMINATED BY AN OUPUT EVENT



INPUT PORT EVENT

A

OUTPUT PORT EVENT

A

ASF INPUT PORT EVENT

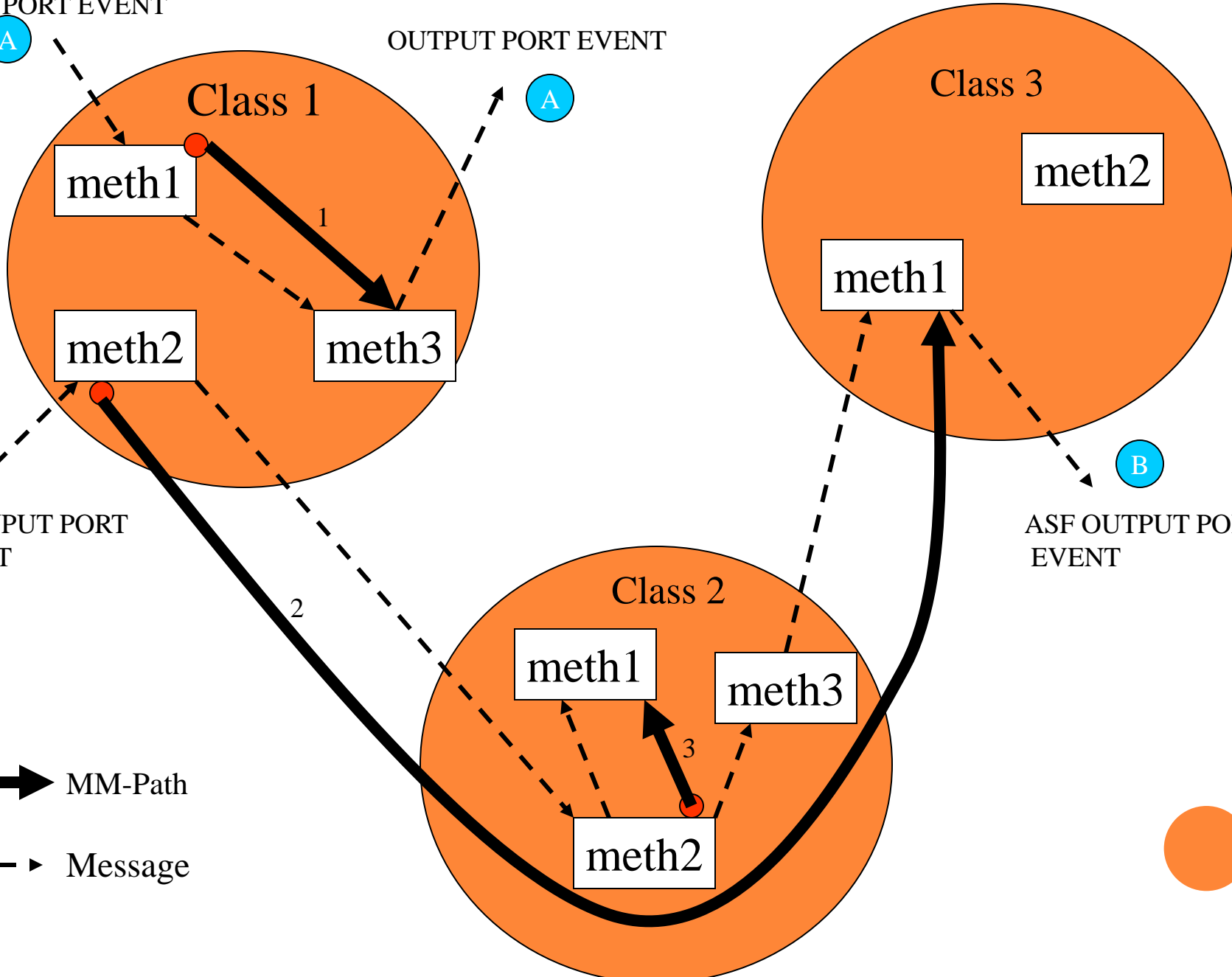
B

ASF OUTPUT PORT EVENT

B

MM-Path

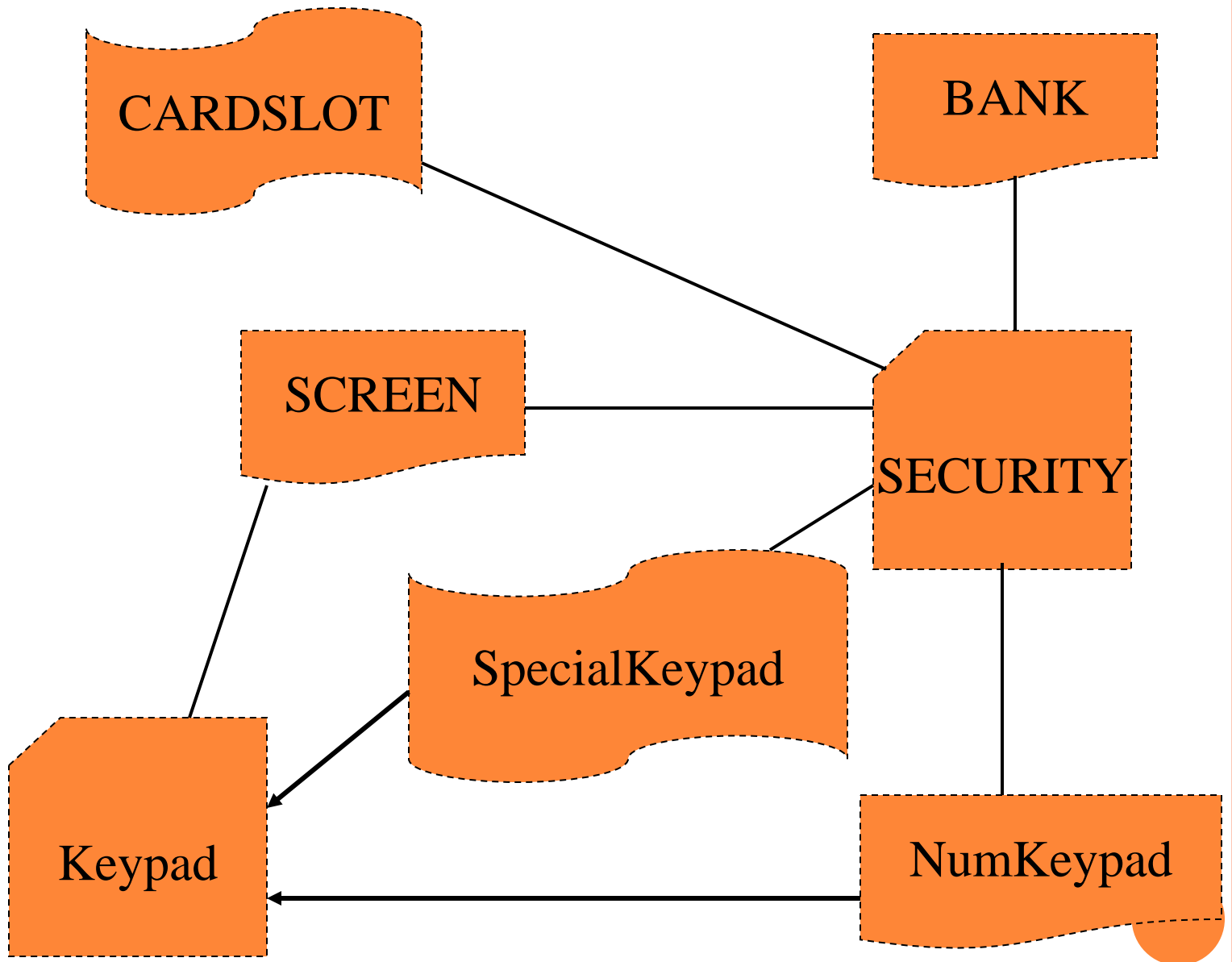
Message

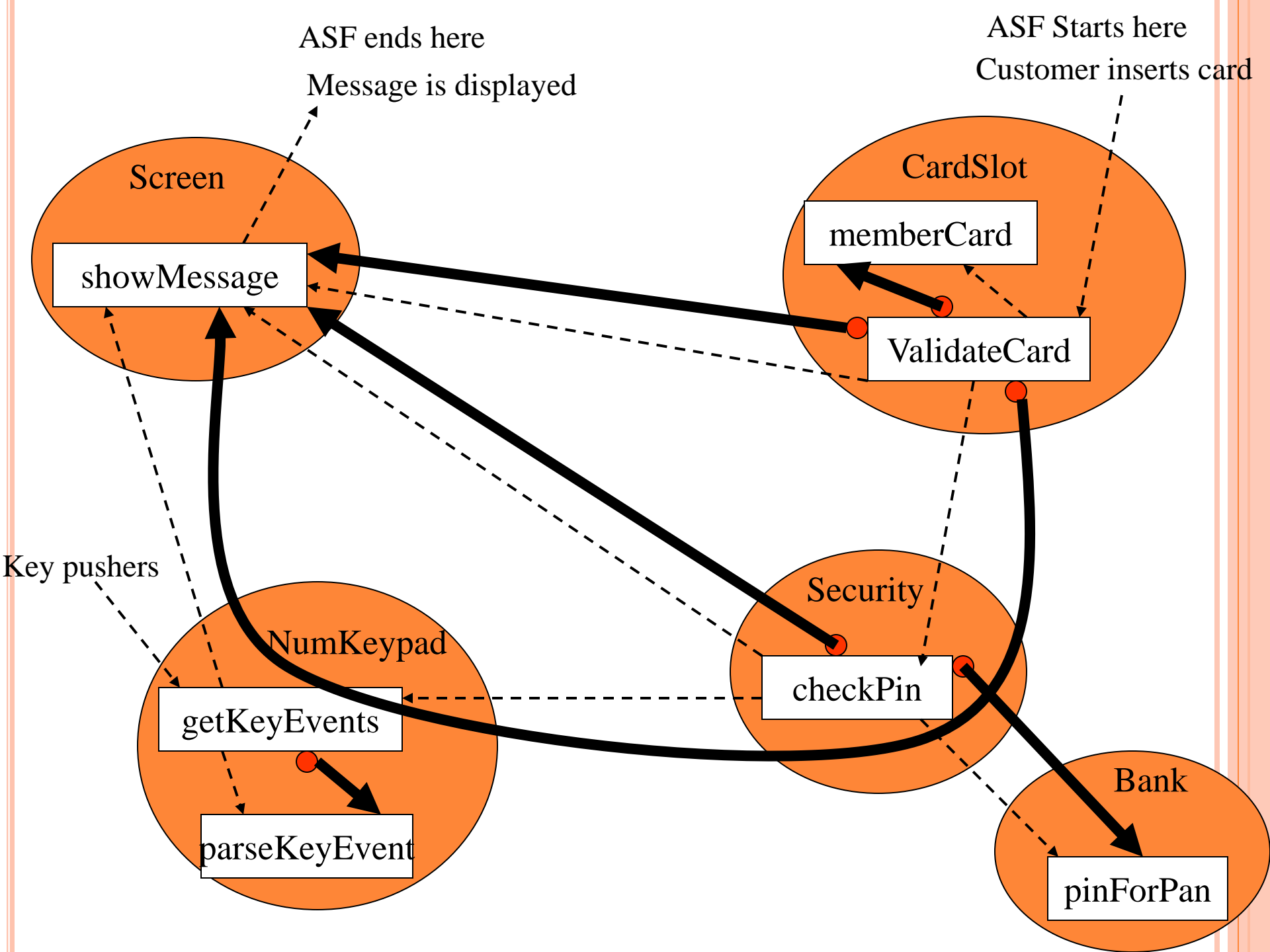


ATM PIN ENTRY

- CUSTOMER ENTERS CARD(EVENT)
- SCREEN REQUESTING PIN ENTRY IS DISPLAYED
- AN INTERLEAVED SEQUENCE OF DIGIT KEY TOUCHES WITH AUDIBLE AND VISUAL FEEDBACK
- POSSIBILITY OF CANCELLATION BY CUSTOMER
- SYSTEM DISPOSITION(VVALID PIN OR CARD RETAINED)







CONCLUSION

- OO TESTING LEVELS- UNIT & SYSTEM SAME AS TRADITIONAL LEVELS
- OO INTEGRATION TESTING IS DIFFERENT AND MORE COMPLEX
- OPTIMAL TEST ORDER SAVES
- TOOLS REQUIRED TO SCALE UP OO TESTING
- LIMIT DESIGNERS TO STRAIGHT INHERITANCE (NO REDEFINING)

