



**Uni Hamburg – Mainframe Summit
z/OS – The Mainframe Operating**

Part 5 – Application Support

Michael Großmann
IBM Technical Sales Mainframe Systems
grossman@de.ibm.com

© Copyright IBM Corporation 2008
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

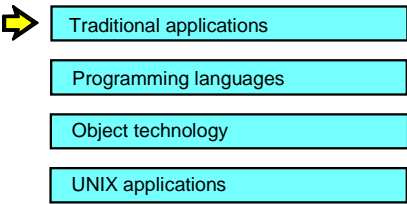
Unit objectives

After completing this unit, you should be able to:

- Describe the support provided for object-oriented application development on z/OS
- Describe the purpose of the Language Environment

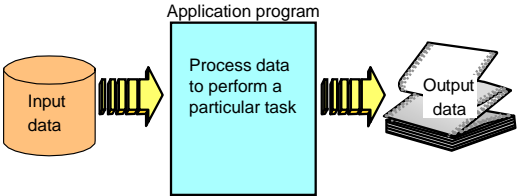
© Copyright IBM Corporation 2008

Traditional applications



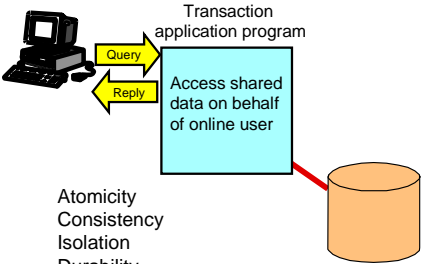
© Copyright IBM Corporation 2008

What is an application?



© Copyright IBM Corporation 2008

What is a transaction?



Atomicity
Consistency
Isolation
Durability

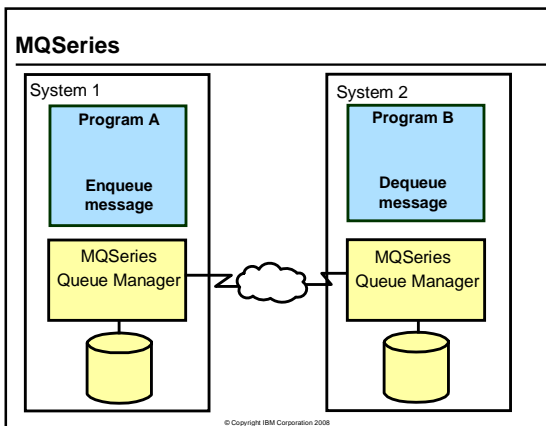
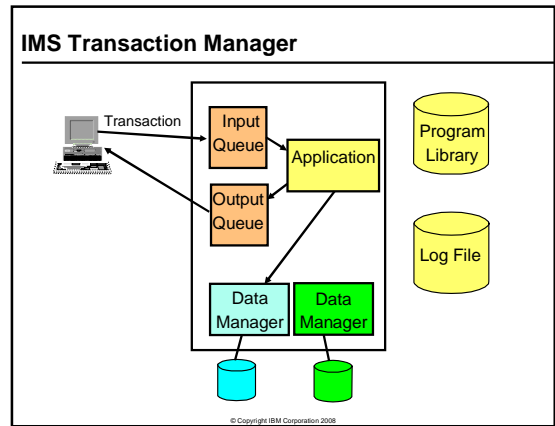
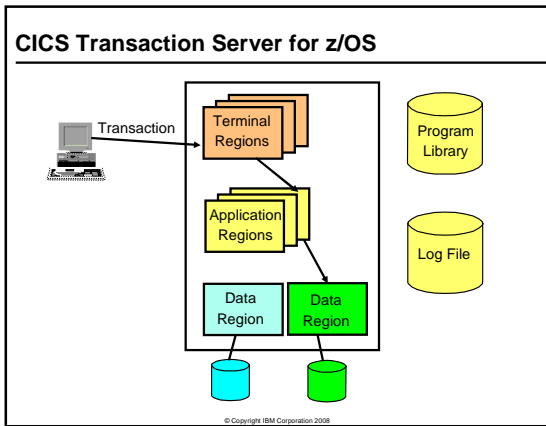
© Copyright IBM Corporation 2008

Transaction processing monitors

- A transaction processing monitor is a program that monitors a transaction as it passes from one stage in a process to another.
- TP monitors ensure that a transaction processes completely, or if an error occurs, that appropriate actions are taken.
- TP monitors connect multiple clients to multiple servers to access multiple databases.

- CICS Customer Information and Control System
- IMS Information Management System
- MQSeries Message Queuing

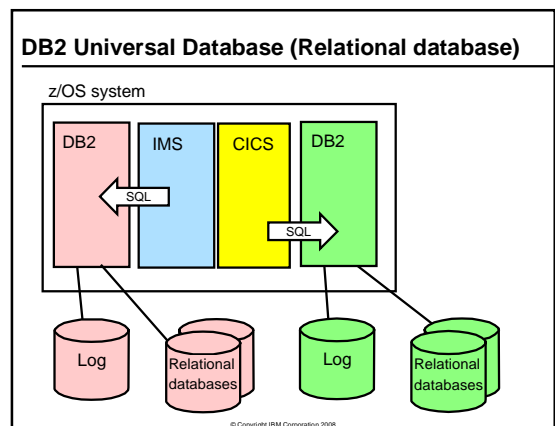
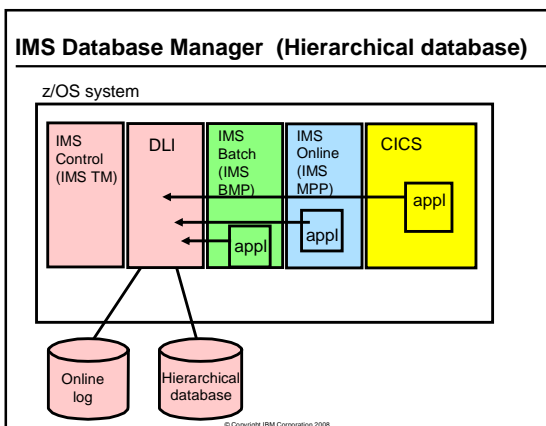
© Copyright IBM Corporation 2008

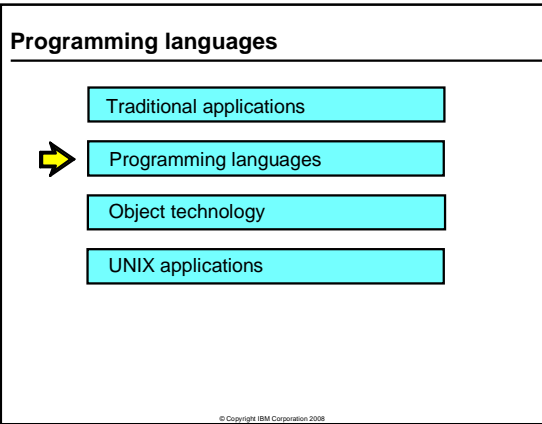


Database management systems

- Database management systems allow multiple independent users to have concurrent access to a central repository of information.
- Database management systems provide data integrity by ensuring that all steps within a unit of work are completed successfully or all changes are backed out.
- Database management systems provide centralized security for the database and allow installations to control which users have access to specific data within the database.

© Copyright IBM Corporation 2008





Traditional programming languages

Assembler

```

XC R_LIST(100),R_LIST      Clear the range list
LD 12,VIRT4_ADDR          Get starting address to pagefix
STG 12,R_START            Save it in range list
LGRH 4,5                  Load number of pages to fix
STG 4,R_PAGES             Save it in range list
SLR 12,12                 Generate primary-space alet
ST 12,R_ALET              Save it in range list
LA 4,R_LIST                Get address of range list
LGRB 4,4                  Make it a 64-bit pointer
STG 4,R_LISTPTR           Save it

```

C/C++

```

/* Now print the string in the file */
fp = fopen("PROG.DAT","rb");
rc = fgets(buffer,sizeof(buffer),fp);
if (rc == NULL)
{
    perror(" Error reading from file ");
    exit(59);
}
printf("%s", buffer);

```

PL/I

```

PUT SKIP LIST( SIG_STRING || SEASON_CODE );
IF RETURN_CODE = 0 THEN
DO; /* Write the signature and its length to the output file. */
FILE_OUT_LINE = SIG_LEN_STRING || SIGNATURE_LENGTH;
WRITE FILE(SIGOUT) FROM( FILE_OUT_LINE );
FILE_OUT_LINE = SIG_STRING || SIGNATURE;
WRITE FILE(SIGOUT) FROM( FILE_OUT_LINE );
END;

```

COBOL

```

PROCEDURE DIVISION.
EXEC CICS READ DATABET(*INFILE*)
REDFLD(INFILE-KEY)
SET ADDRESS OF VAR-RECORD
LENGTH(RECORD-LEN)
END-EXEC
MOVE VAR-RECORD TO MS-RECORD-HOLD.

```

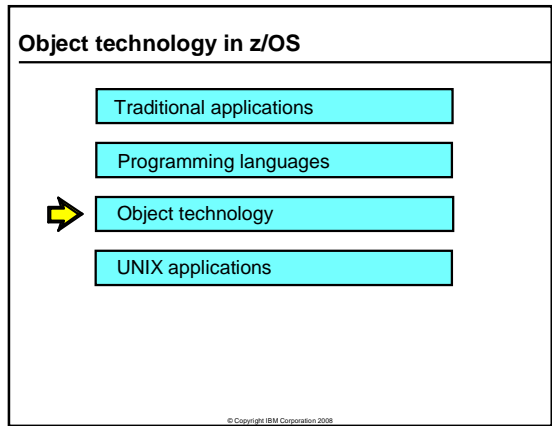
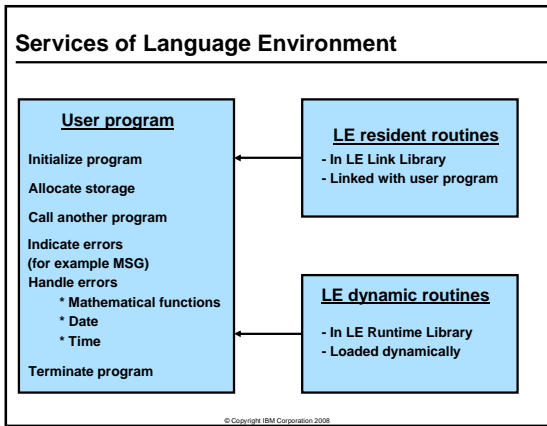
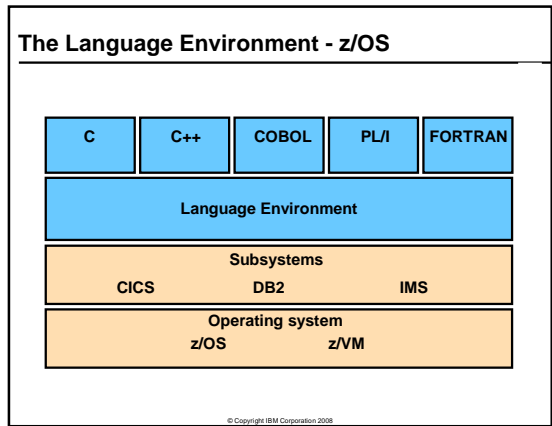
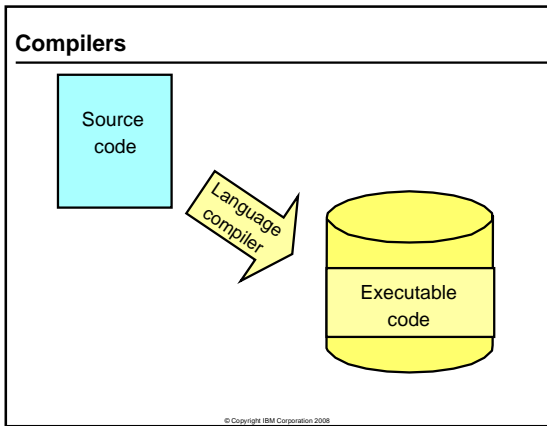
FORTRAN

```

IF (MOLA_CTOK(1:8) .EQ. FOR1003) THEN
  INPUT_FILE_PTR = QDQOC(MOLA_CTOK, 6)
  INPUT_FILEID(QDINF4(MOLA_CTOK, 7)) = '0'
  FIXUP_CTOK(1:8) = CNDICE
  FIXUP_CTOK(9:12) = Z'00000000'
  RESPONSE = 60
  ENDDIF

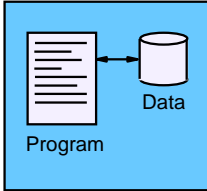
```

© Copyright IBM Corporation 2008

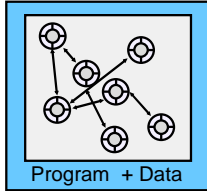


What is object technology?

Conventional programming

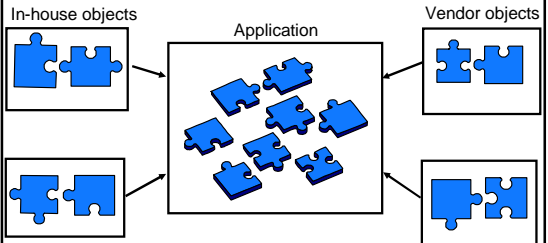


Object-oriented programming



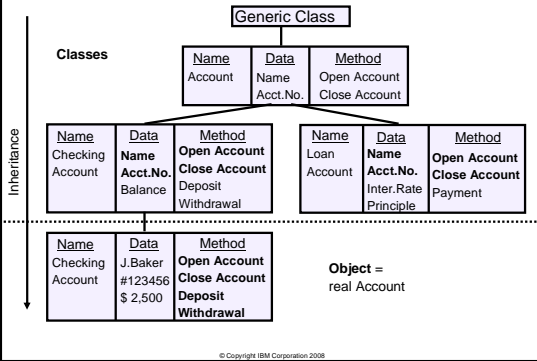
© Copyright IBM Corporation 2008

Object-oriented technology



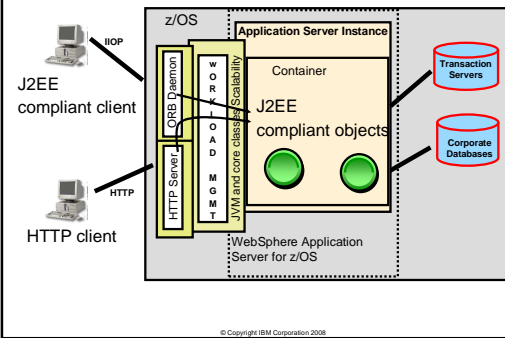
© Copyright IBM Corporation 2008

Classes and objects



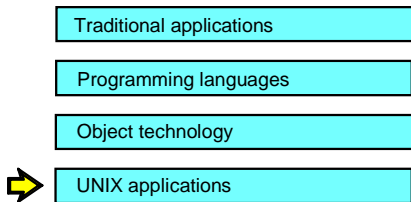
© Copyright IBM Corporation 2008

J2EE and Enterprise Java Beans



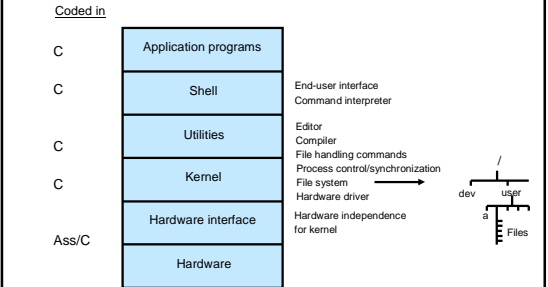
© Copyright IBM Corporation 2008

UNIX applications



© Copyright IBM Corporation 2008

UNIX basic structure



© Copyright IBM Corporation 2008

