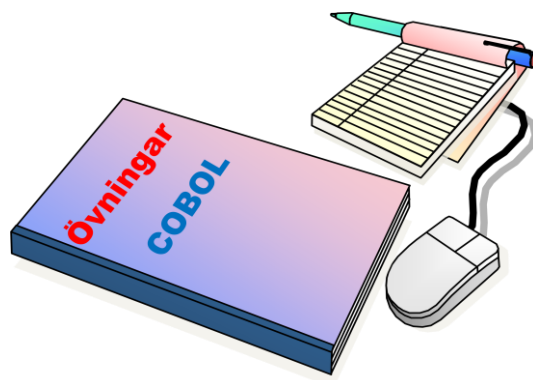


Övningar

V25.1



Innehållsförteckning

Övning 1 - Kompilering och Länkning.....	4
Uppgift 1 - Ett enkelt program	4
Övning 2 - Fella.....	5
Uppgift 1 - Komplettering.....	5
Övning 3 - Data Division och Working-Storage Section.....	6
Uppgift 1 - Working-Storage Section	6
Övning 4 - Gruppbeskrivningar	7
Uppgift 1 - Gruppbeskrivning	7
Uppgift 2 - Flytta variabel.....	8
Övning 5 - Redigering.....	9
Uppgift 1 - Utskrift.....	9
Övning 6 - Aritmetik.....	10
Uppgift 1 - Faktura.....	10
Övning 7 - Strängar.....	12
Uppgift 1 - String	12
Övning 8 - Perform	13
Uppgift 1 - Out-Of-Line.....	13
Uppgift 2 – Perform times.....	13
Uppgift 3 - Räknare.....	14
Uppgift 4 – Perform Varying	14
Uppgift 5 - In-Line-Perform.....	14
Övning 9 - Utvärdering.....	15
Uppgift 1 – Utvärdering med If.....	15
Uppgift 2 – Villkorsvariabel	15
Uppgift 3 – Utvärdering med Evaluate	16
Övning 10 - Klasstest.....	17
Uppgift 1 – Analys	17
Övning 11 – Tabeller och Redefines	18
Uppgift 1 - Sammansatt utskrift	18
Övning 12 - Open, Read.....	19
Uppgift 1 - Display av filinnehåll	19
Övning 13 - Artikelregister.....	20
Uppgift 1 - Enkel utskrift	20
Uppgift 2 - Redigerad utskrift	21
Övning 14 - Redigerad utskrift	22
Uppgift 1 - Radsummor.....	22
Uppgift 2 - Totalsummor.....	22
Övning 15 - Sidbrytning	24
Uppgift 1–Sidbrytning, sidstorlek	24
Uppgift 2 - Sidbrytning, artikelgrupp.....	24

Övning 16 - Produkt-beskrivning	25
Uppgift 1 - VSAM KSDS	25
Övning 17 - Tabellsökning.....	26
Uppgift 1 - Fyll produkttabell	26
Övning 18 - Subprogramanrop.....	27
Uppgift 1 - Fyll tabell i subprogram.....	27
Övning 19 - Inbyggda funktioner	28
Uppgift 1 - Bäst före datum.....	28
Övning 20 - Driverprogram	30
Uppgift 1 - Exekveringsparameter	30
Övning 21 - Indatakontroll.....	31
Uppgift 1 - Numerisk test.....	31
Övning 22 - Filmatchning.....	32
Uppgift 1 - Ändringstransaktioner	32
Övning 23 - Lagringsformer.....	34
Uppgift 1 – Skapa fil	34
Uppgift 2 – Skriv program	34

Övning 1 - Kompilering och Länkning

I denna övning så skall du bekanta dig med utvecklingsmiljön, samt se hur du gör kompilering och länkning.

Uppgift 1 - Ett enkelt program

- Du skall skriva ett minimalt program med nödvändiga Divisions i Cobol.
- Programnamn kan du välja själv, med det är praktiskt att du använder övningens nummer i kombination med uppgiftens nummer t.ex. OVN11.

```
Identification Division.  
Program-Id. OVN11.  
*-----  
* Detta är mitt första COBOL-program  
*-----  
Procedure Division.  
Avsluta.  
  GoBack  
  .
```

- Kompilera programmet.
- När kompileringen är felfri, så kontaktar du kursledaren, så får du information om hur du kan exekvera/testa programmet.
- Exekvera programmet och kontrollera resultatet.

Uppgiften är avslutad.

Övningen är avslutad.

Övning 2 - Fellista

I denna övning kommer du att komplettera programmet med några uppgifter. Målet är att du skall få ett kompileringsfel, så att du lär dig principerna vid hanteringen av dessa fel.

Uppgift 1 - Komplettering

- Kopiera programmet från Övning 1 och ge det lämpligt namn.
- Komplettera programmet med följande uttryck i Procedure Division innan uttrycket GoBack. Notera den avsiktliga felstavningen av Display.

```
Diplay 'Detta är mitt första program'
```

- Kompilera programmet.
- Kontrollera fellistan.
- Korrigera felet och kompilera programmet igen.
- När kompileringen är felfri, så kontaktar du kursledaren, så får du information hur du kan kontrollera resultatet.

Uppgiften är avslutad

Övningen är avslutad.

Övning 3 - Data Division och Working-Storage Section

I denna övning kommer du att komplettera det tidigare programmet med beskrivningar i Working-Storage Section, som skall finnas i Data Division.

Uppgift 1 - Working-Storage Section

- Kopiera programmet från Övning2.
- Komplettera programmet med följande uttryck i Working-Storage Section:

```
Data Division.  
Working-Storage Section.  
01 Meddelande    Pic X(35) Value 'Detta är COBOL'.
```

- Komplettera även Procedure Division så att du gör Display av den skapade Meddelande-variabeln.
- Kompilera programmet och kontrollera resultatet.
- Exekvera programmet och kontrollera resultatet.

Uppgiften är avslutad.

Övningen är avslutad.

Övning 4 - Gruppbeskrivningar

I denna övning kommer du att komplettera det tidigare programmet med ytterligare beskrivningar i Working-Storage Section.

Uppgift 1 - Gruppbeskrivning

- Kopiera programmet från Övning 3.
- Programmet skall nu kompletteras med följande uttryck i Working-Storage Section.

```
Working-Storage Section.  
01 Meddelande.  
    05          Pic X(08) Value 'Detta är'.  
    05 Datat    Pic X(06).  
    05          Pic X(04) Value 'övning 4'.  
Procedure Division.  
    Move 'COBOL' to Datat  
    Display Meddelande  
    GoBack  
    .
```

- Notera den avsiktliga felaktigheten i definitionen `Pic X(04)`.
- Kompilera programmet och kontrollera resultatet.
- Korrigera och kompilera tills du har en felfri kompilering.
- Exekvera programmet och kontrollera resultatet.
- Korrigera och kompilera igen om meddelandet inte har detta utseende:

```
Detta är COBOL övning 4
```

Uppgiften är avslutad.

Uppgift 2 - Flytta variabel

- Komplettera programmet så att värdet 'COBOL' finns i en variabel på 77-nivå.
- Flytta in innehållet i denna variabel till variabeln med namnet `Data1`.
- Kompilera och korrigera tills du har en felfri kompilering
- Exekvera programmet och kontrollera resultatet.

Uppgiften är avslutad.

Övningen är avslutad.

Övning 5 - Redigering

I denna övning så skall du bekanta dig med de redigeringskonstanter som finns i COBOL för att redigera en utskrift.

Uppgift 1 - Utskrift

- Programmet skall skapa ett antal värde-variabler (*numeric*) som innehåller numeriska värden. Vilka värden de skall innehålla ser du i bilden nedan.
- Beskriv olika rader som är grupper (*group item*) på 01-nivå, som innehåller både texter (*alphanumeric*) samt variabler med redigeringskonstanter (*numeric-edited*) på lägre nivå.
- Flytta sedan en numeriska värde-variabel till variabeln på en rad. Gör detta för varje rad. När du sedan gör utskrift (*Display*) av raderna, så skall resultatet bli som du ser på nedanstående bild.

```
*-----*
* Detta är en utskrift från Grupp xx
*-----*
Utan nollundertryckning  00001234
Med nollundertryckning   1234
Med utfyllnad            *****456
Positivt tal med tecken  +   1,25
Negativt tal med tecken -   1,25
Positivt tal utan tecken    4,50
Negativt tal med tecken   -4,50
Negativt tal med tecken    7,25-
*-----*
* Slut på utskrift
*-----*
```

- Kompilera och korrigera tills du har en felfri kompilering.
- Exekvera programmet och kontrollera resultatet.

Om du har tid: Prova gärna några fler alternativ som du sett exempel på i kursboken.

Uppgiften är avslutad.

Övningen är avslutad.

Övning 6 - Aritmetik

Nu har vi kommit fram till övningar med aritmetik, där du får prova på att utföra aritmetik med flera olika COBOL-verb som bas.

Uppgift 1 - Faktura

- Du skall skapa ett nytt program som skall göra utskrift med `Display`, så att resultatet liknar det du ser i nedanstående bild.
- Börja med att göra beräkningen på papper så ser du vilka värde- och resultat-variabler du kommer att behöva. Tänk också igenom vilka namn du vill ha på dessa variabler.
- Skapa ett antal numeriska värde-variabler med de olika värdena som du behöver, även antal, rabatt- och moms-procent.
- Beskriv olika rader som är grupper, som innehåller både text samt variabler med plats för numeriska värden.
- Gör nödvändiga beräkningar med de numeriska värde-variablerna och spara resultatet i numeriska värde-variabler.
- Flytta sedan en numerisk värde/resultat-variabel till en redigerad variabel på en rad på samma sätt som i föregående övning.
- Du skall ha ett resultat som liknar det på nedanstående bild när du gör `Display` av raderna.

Fakturaspecifikation		
01	Stänkskärm vä-fram, a 1250:-	nnnn:-
03	Vindrutetorkare, a 390:-	nnnn:-
	Delsumma	nnnnnn:-
	Avgår rabatt 15%	nnnn:-
	Att betala exkl moms	nnnnnn:-
	Tillkommer moms 25%	nnn:-
	Totalsumma att betala	SEK nnnnnn:-
Slut på Fakturaspecifikation		

- Kompilera och korrigera tills du har en felfri kompilering.
- Exekvera programmet och kontrollera resultatet.

Om du har tid: Prova gärna att använda decimaltal vid beräkningarna samt använd olika värden i antalsvariablerna för de olika produkterna.

Uppgiften är avslutad.

Övningen är avslutad.

Övning 7 - Strängar

Nu har tiden kommit för att prova på att skapa olika strängar från andra befintliga strängar, och du kommer bl.a. använda COBOL-verbet String.

Uppgift 1 - String

- Kopiera programmet från Övning 4.
- Skapa en ny variabel med namnet `Nyttmeddelande` med längden 80 tecken. Skulle du behöva en större variabel kan du komplettera detta senare.
- Skapa två (2) nya variabler med namnet `Fnamn` respektive `Enamn`. De skall vara 20 tecken långa och innehålla ditt förnamn respektive efternamn.
- Hämta in dagens datum med verbet `Accept`. Kontrollera i COBOL LanguageReference-manualen på nätet hur uttrycket skall se ut.
- Du skall nu skapa innehållet till variabeln `Nyttmeddelande`. Detta skall bestå av samtliga element från strukturen `Meddelande`, som redan fanns i ditt program. Datat från de ingående elementen, och andra, skall separeras med blanktecken. Utöver detta så skall dagens datum i formatet `dd/mm`, artal finnas med. Slutligen skall du ha med texten 'hälsningar från *Bo Ek*'. Ersätt med data från variablerna `Fnamn` samt `Enamn`.
- Kompilera och korrigerar tills du har en felfri kompilering.
- Exekvera programmet och kontrollera resultatet.

Uppgiften är avslutad.

Övningen är avslutad.

Övning 8 - Perform

I denna övning kommer du att använda olika Perform-konstruktioner för att utföra kod i ditt program.

Uppgift 1 - Out-Of-Line

- Skriv ett program som ser ut som följer.

```
Working-StorageSection.  
01 Meddelande    Pic X(25) Value 'Ett Meddelande'.  
  
ProcedureDivision.  
    Perform Skrivmeddelande  
    GoBack  
.  
Skrivmeddelande.  
    Display Meddelande  
.
```

- När du har gjort detta så har du skapat ett program som har en paragraf med namnet Skrivmeddelande Detta utförs med en Out-Of-Line perform-sats.
- Kompilera och korrigerar tills du har en felfri kompilering.
- Exekvera programmet och kontrollera resultatet.

Uppgiften är avslutad.

Uppgift 2 – Perform times

- Kopiera det tidigare programmet och komplettera så att paragrafen Skrivmeddelande utförs 10 gånger.
- Kompilera och korrigerar tills du har en felfri kompilering.
- Exekvera programmet och kontrollera resultatet.

Uppgiften är avslutad.

Uppgift 3 - Räknare

- Kopiera det tidigare programmet och komplettera så att paragrafen `Skrivmeddelande` även skriver ut följande:

```
Det är gång nr nn
```

- Kompilera och korrigera tills du har en felfri kompilering.
- Exekvera programmet och kontrollera resultatet.

Uppgiften är avslutad.

Uppgift 4 – Perform Varying

- Kopiera det tidigare programmet och komplettera så att paragrafen `Skrivmeddelande` skrivs ut enligt Uppgift 2, men använd konstruktionen `Varying` i `Perform`-uttrycket, om du inte redan gjort detta.
- Kompilera och korrigera tills du har en felfri kompilering.
- Exekvera programmet och kontrollera resultatet.

Uppgiften är avslutad.

Uppgift 5 - In-Line-Perform

- Kopiera det tidigare programmet och korrigera så att du använder en `In-Line-Perform`-konstruktion i stället för `Out-Of-Line` som du har använt hittills.
- Kompilera och korrigera tills du har en felfri kompilering.
- Exekvera programmet och kontrollera resultatet.

Uppgiften är avslutad.

Övningen är avslutad.

Övning 9 - Utvärdering

I denna övning kommer du att prova på att använda konstruktionen If Else End-If samt Evaluate

Uppgift 1 – Utvärdering med If

- Kopiera programmet som du använde i Uppgift 3 i föregående övning.
- Du skall komplettera programmet så att det gör följande Display:

```
Det är färre än fem gånger  
Det är fler än eller fem gånger  
Det är tionde gången
```

- Notera att flera förekomster av samma meddelande kan förekomma i utskriften.
- Du skall heller inte använda sammansatta villkor och negationer.
- Kompilera och korrigera tills du har en felfri kompilering.
- Exekvera programmet och kontrollera resultatet.

Uppgiften är avslutad.

Uppgift 2 – Villkorsvariabel

- Komplettera programmet så att en villkorsvariabel används för att indikera när maxantalet 10 är uppnått.
- Kompilera och korrigera tills du har en felfri kompilering.
- Exekvera programmet och kontrollera resultatet.

Uppgiften är avslutad.

Uppgift 3 – Utvärdering med Evaluate

- Kopiera programmet som du använde i Uppgift 2 i Övning 8.
- Programmet skall kompletteras så att det gör följande `Display`:

```
Det är andra gången  
Det är fjärde gången  
Det är sjätte gången  
Det är åttonde gången  
Det är tionde gången
```

- För alla övriga tillfällen så skall följande utskrift göras:

```
Det är en udda gång
```

- Kompilera och korrigera tills du har en felfri kompilering.
- Exekvera programmet och kontrollera resultatet.

Uppgiften är avslutad.

Övningen är avslutad.

Övning 10 - Klasstest

Detta program kommer att analysera innehållet i en textvariabel och skriva ut meddelande.

Uppgift 1 – Analys

- Programmet OVN101 skall läsa in en textsträng med COBOL-verbet ACCEPT. Läs i manualen hur det skall se ut.
- Placera textsträngen som instream-data i ditt exekveringsjobb COBEXEC. Detta kan se ut så här:

```
//UTB00X    JOB NOTIFY=&SYSUID,TIME=(,1)
//JOB LIB    DD DSN=UTB00.AP15.LOADLIB,DISP=SHR
//STEG1     EXEC PGM=X1
//SYSIN      DD *
DETTA ÄR EN STRÄNG TILL EN ÖVNING I COBOL-KURSEN AP15
```

- Den alfanumeriska variabeln i Working-Storage som skall ta emot strängen, skall beskrivas som Pic X(72) .
- Därefter skall strängen undersökas och följande Display-meddelande skall skrivas ut:

```
*** Textvariabelns längd      : 72
*** Textvariabelns innehåll   : DETTA ÄR EN STRÄNG T. .
*** Totalt antal tecken      : 57
*** Totalt antal siffror     : 02
*** Totalt antal vokaler     : 16
*** Totalt antal konsonanter : 29
*** Antal andra tecken       : 10
*** Andra tecken             : -
```

- Kompilera och korrigera tills du har en felfri kompilering.
- Exekvera programmet och kontrollera resultatet.

Uppgiften är avslutad.

Övningen är avslutad.

Övning 11 – Tabeller och Redefines

Som du förstår av rubriken så kommer du att använda tabellkonstruktioner samt redefines för att fylla en tabell med data. Därefter skall du göra en utskrift.

Uppgift 1 - Sammansatt utskrift

- Skapa ett nytt program.
- Programmet skall hämta datum samt aktuell veckodag med COBOL-verbet `ACCEPT`. Du har ju redan använt det så du vet hur du läser mer om det.
- Programmet skall göra en utskrift som liknar följande:

```
I dag är det dagen den nn manaden och klockan är hh:mm
```

- Kompilera och korrigera tills du har en felfri kompilering.
- Exekvera programmet och kontrollera resultatet.

Uppgiften är avslutad.

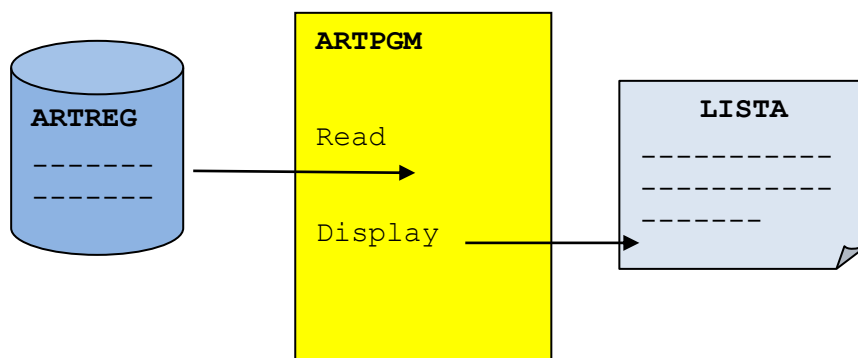
Övningen är avslutad.

Övning 12 - Open, Read

Nu har vi kommit fram till att arbeta med filer. I denna övning så kommer du att arbeta med en sekvensfil, som du skall läsa och sedan visa innehållet imed Display.

Uppgift 1 - Display av filinnehåll

- Filen som du skall läsa har en fast postlängd av 80 tecken. Du kan använda valfritt namn för ditt namn i programmet, och även i `Assign to` uttrycket. Maximalt 8 tecken i `Assign`. Innan du testar programmet så måste du korrigera din exekverings-JCL.
- Skapa ett nytt program och gör beskrivning för att läsa en sekvens-fil med fast postlängd av 80 tecken. Visa varje läst rad med COBOL-verbet `Display`. Du skall även kontrollera filstatus efter `Open` och skriva nödvändiga meddelanden.



- Kompilera och korrigera tills du har en felfri kompilering.
- Exekvera programmet och kontrollera resultatet.

Uppgiften är avslutad.

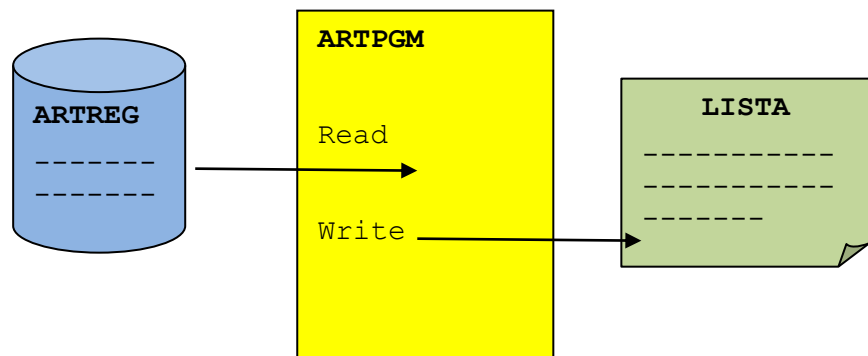
Övningen är avslutad.

Övning 13 - Artikelregister

Nu kommer du att titta närmare på innehållet i de lästa posterna samt skriva en lista. Du kommer inte att göra någon redigering vid utskriften i Uppgift 1, utan gör i princip som i föregående övning, fast du kommer att skriva till en fil i stället för att använda verbet Display. I uppgift 2 kommer du att redigera de skrivna raderna.

Uppgift 1 - Enkel utskrift

- Kopiera det föregående programmet.
- Programmet skall läsa från artikelregistret och skriva till en lista. Listan kan du betrakta som en sekventiell fil. Vilka interna och externa namn du använder saknar betydelse, det viktiga är bara att du kompletterar din JCL innan exekveringarna, för att matcha de krav du beskrivit i ditt program.



- Posterna har följande utseende:

Beskrivning	Längd	Typ
Artikelnummer	5	N
Filler	1	
Benämning	25	A
Filler	1	
Artikeltyp	1	A
Filler	1	
Lagerplats1	2	A
Kvantitet1	6	N
Lagerplats2	2	A
Kvantitet2	6	N
Lagerplats3	2	A
Kvantitet3	6	N
Filler	22	

(A = Alfnumerisk, N= Numerisk)

- Kompilera och korrigera tills du har en felfri kompilering.
- Exekvera programmet och kontrollera resultatet.

Uppgiften är avslutad.

Övningen är avslutad.

Uppgift 2 - Redigerad utskrift

- Komplettera programmet så att utskriften av de artikelraderna är redigerad, d.v.s. raden innehåller blankutfyllnad så att de olika fälten kan urskiljas.
- Du skall även skriva en rubrikrad, som anger vad som finns i artikelradernas olika kolumner. Dessutom så skall du skriva en avslutningsrad sist i listan.
- Listan skall ha ett utseende som påminner om denna:

Utskrift av Artikelregister							
Artnr	Artnamn	Lp1	Antal	Lp2	Antal	Lp3	Antal
12345	Artikel1	6	12	.	.	.	
Slut på utskrift							

- Kompilera och korrigera tills du har en felfri kompilering.
- Exekvera programmet och kontrollera resultatet.

Uppgiften är avslutad.

Övningen är avslutad.

Övning 14 - Redigerad utskrift

Nu kommer programmet att kompletteras med lite ytterligare logik, för att göra diverse summeringar, så att det kan göra anspråk att vara ett "riktigt" program.

Uppgift 1 - Radsummer

- Kopiera programmet från föregående uppgift.
- Programmet skall kompletteras så att det gör summering på radnivå, d.v.s. du skall summera antalet artiklar på samtliga lagerplatser för ett visst artikelnummer och skriva denna summa för varje artikelrad.

Artnr	Artnamn	Lp1	Antal	Lp2	Antal	Lp3	Antal	Summa
12345	Abc	n	12	n	23	n	16	51

- Kompilera och korrigera tills du har en felfri kompilering.
- Exekvera programmet och kontrollera resultatet.

Uppgiften är avslutad.

Uppgift 2 - Totalsummer

- Kopiera programmet från föregående uppgift.
- Programmet skall kompletteras så att det gör summering på lagerplatsnivå, d.v.s. du skall summera antalet artiklar för varje lagerplats och redovisa detta i slutet av rapporten på en egen rad.

Artnr	Artnamn	Lp1	Antal	Lp2	Antal	Lp3	Antal	Summa
12345	Abc	n	12	n	23	n	16	51
Totalsummor			x1		x2		x3	

- Kompilera och korrigerar tills du har en felfri kompilering.
- Exekvera programmet och kontrollera resultatet.

Uppgiften är avslutad.

Övningen är avslutad.

Övning 15 - Sidbrytning

Du kommer nu att komplettera ditt program med ytterliga logik för att göra sidbrytningar på olika ställen.

Uppgift 1–Sidbrytning, sidstorlek

- Kopiera programmet från föregående uppgift.
- Programmet skall nu kompletteras så att det gör en sidbrytning efter ett visst antal rader. Det skall finnas plats för 25 rader per sida. Varje sida skall innehålla rubrik.
- Kompilera och korrigera tills du har en felfri kompilering.
- Exekvera programmet och kontrollera resultatet.

Uppgiften är avslutad.

Uppgift 2 - Sidbrytning, artikelgrupp

- Kopiera programmet från föregående uppgift.
- Programmet skall nu kompletteras så att det gör en sidbrytning för varje artikelgrupp. Artikelgruppen utgörs av första siffran i artikelnumret.
- Summeringar skall göras på varje sida, för varje artikelgrupp.
- Kompilera och korrigera tills du har en felfri kompilering.
- Exekvera programmet och kontrollera resultatet.

Uppgiften är avslutad.

Övningen är avslutad.

Övning 16 - Produktbeskrivning

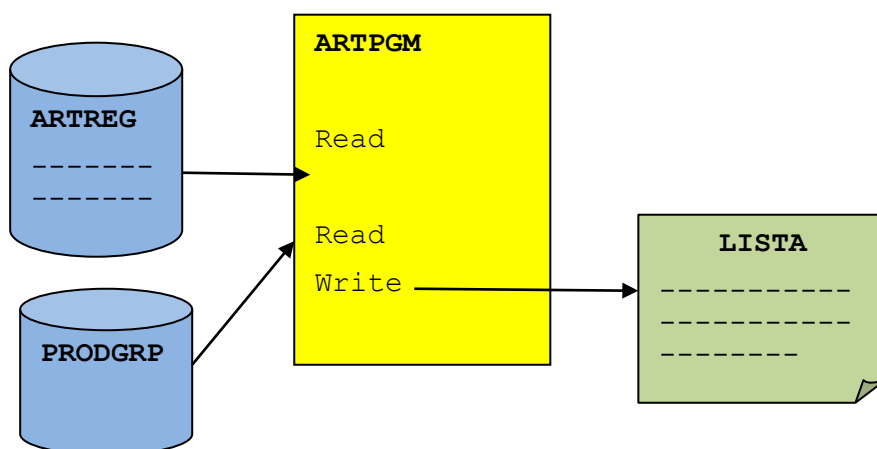
Nu skall du komplettera listan, så att varje sida, som ju nu innehåller information om en viss produktgrupp, även skall innehålla en beskrivning av produktgruppen. Denna beskrivning finns i en indexerad fil där produktnumret är nyckelvärde.

Uppgift 1 - VSAM KSDS

- Kopiera programmet från föregående uppgift.
- Programmet skall för varje ny produktgrupp, skriva ut en produktbeskrivning på varje ny sida/produktgrupp. Denna beskrivning finns i en VSAM-KSDS-fil, där produktgruppen är nyckelvärde, och har följande utseende:

Postlängd: 80

<u>Beskrivning</u>	<u>Längd</u>	<u>Typ</u>
Nyckel	1	X
Beskrivning	79	X



- Kontakta din kursledare när du är klar att testa ditt program, så att du kan komplettera din JCL för exekveringen.
- Kompilera och korrigera tills du har en felfri kompilering.
- Exekvera programmet och kontrollera resultatet.

Uppgiften är avslutad.

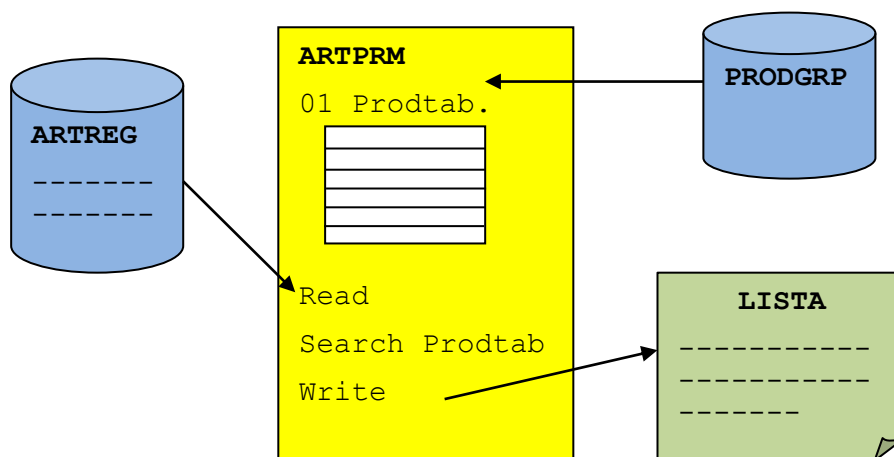
Övningen är avslutad.

Övning 17 - Tabellsökning

Du kommer nu att läsa in samtliga poster från filen med produktbeskrivning och placera dessa i en intern COBOL-tabell. Härifrån kommer du sedan att hämta beskrivningen när du behöver den, i stället för att läsa filen vid varje ny produktgrupp.

Uppgift 1 - Fyll produkttabell

- Kopiera programmet från föregående uppgift.
- Komplettera programmet så att det läser samtliga produktbeskrivningar från den indexerade filen som du använde i föregående övning och sparar i en intern tabell.
- Hämta sedan produktbeskrivningen från denna interna tabell i stället för att läsa den indexerade filen.



- Kompilera och korrigera tills du har en felfri kompilering.
- Exekvera programmet och kontrollera resultatet.

Uppgiften är avslutad.

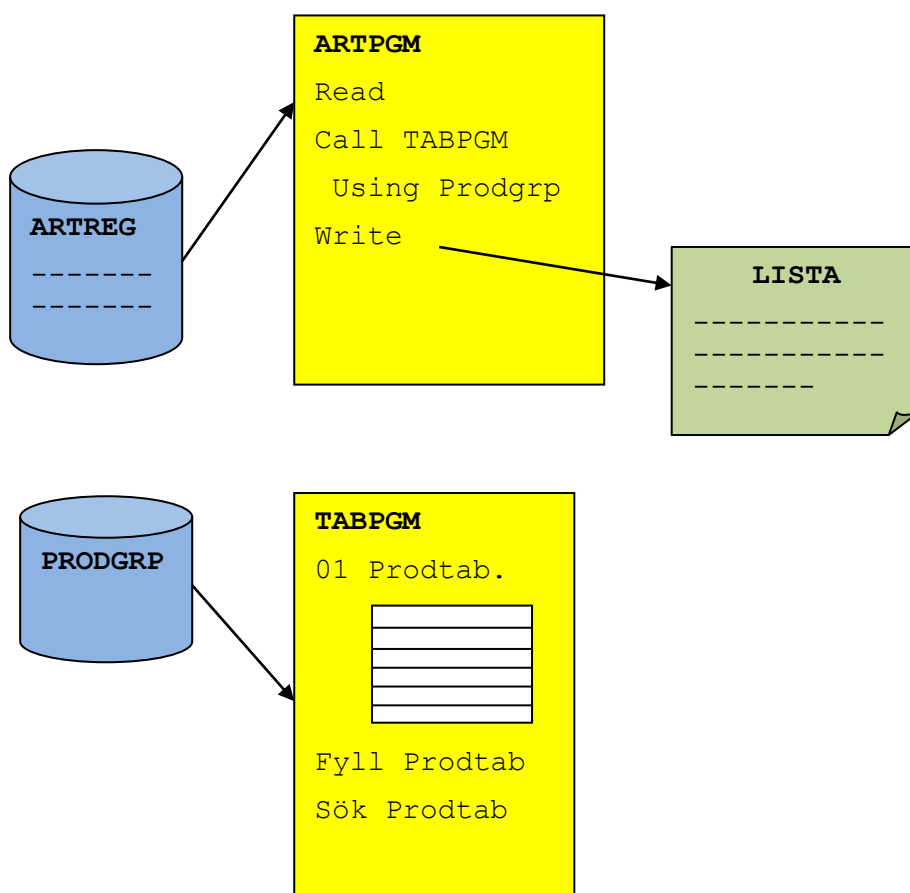
Övningen är avslutad.

Övning 18 - Subprogramanrop

Nu kommer du att flytta ut koden som fyller den interna COBOL-tabellen med produktbeskrivningar, till ett eget subprogram.

Uppgift 1 - Fyll tabell i subprogram

- Kopiera programmet från föregående uppgift.
- Du skall nu dela upp detta program i två (2) program, där det nya programmet skall ansvara både för att fylla tabellen med produktbeskrivningarna samt söka tabellen med angivet prodgruppnummer.



- Kompilera och korrigera tills du har en felfri kompilering.
- Exekvera programmet och kontrollera resultatet.

Uppgiften är avslutad.

Övningen är avslutad.

Övning 19 - Inbyggda funktioner

Äntligen får du prova på att använda några inbyggda funktioner.

Uppgift 1 - Bäst före datum

- Kopiera huvudprogrammet från föregående övning.
- Programmet skall nu kompletterasså att följande information finns tillgänglig i sidhuvudet på första sidan, som informerar om dagens datum samt rapportens giltighetstid, som är 30 dagar från dagens datum:

Utskrift av Artikelregister

Utskrivet: den n manadsnamnx 20xx

Giltig t.o.m : den n manadsnamny 20xx

Artnr	Artnamn	Lp1	Antal	Lp2	Antal	Lp3	Antal	Summa
-------	---------	-----	-------	-----	-------	-----	-------	-------

- För att skapa de olika datumen, skall du komplettera programmet med ett inbyggt program, som skall tillhandahålla de olika datumen
- Det inbyggda programmet kan ha namnet `BastForeDatum`.

Program-Id. ARTPGM.

Call "BastForeDatum" Using ...

Program-Id. BastForeDatum.

End Program Bastforedatum.

End Program ARTPGM.

- Kompilera och korrigera tills du har en felfri kompilering.
- Exekvera programmet och kontrollera resultatet.

Uppgiften är avslutad.

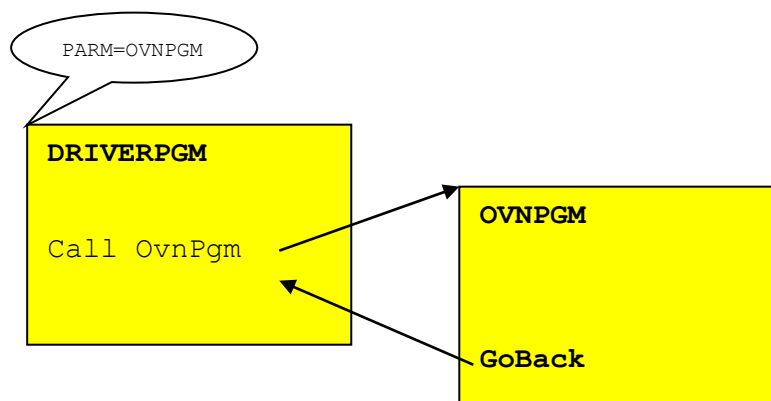
Övningen är avslutad.

Övning 20 - Driverprogram

I denna övning kommer du att skriva ett program som fungerar som ett driverprogram och skall med ledning av en exekveringsparameter anropa valt program.

Uppgift 1 - Exekveringsparameter

- Skapa ett nytt program.
- Programmet skall ta emot en exekveringsparameter, som skall vara namnet på ett av dina övningsprogram, och skall med ledning av denna parameter anropa valt program.
- Kontrollera att anropet fungerar som det är tänkt. Vid en avvikelse så skall du avsluta programmet med ett felmeddelande i jobbloggen, samt returkod 12.



- Kompilera och korrigera tills du har en felfri kompilering.
- Exekvera programmet med olika parametrar och kontrollera resultatet.

Uppgiften är avslutad.

Övningen är avslutad

Övning 21 - Indatakontroll

I denna övning så skall du kontrollera innehållet i artikelregistret så att det uppfyller ställda krav.

Uppgift 1 - Numerisk test

- Komplettera programmet från Övning 16.
- Programmet skall kontrollera att antalet poster i varje lagerplats är numerisk.
- När du upptäcker ett fel så skall du skriva en log som du först måste skapa under TSO/ISPF.
- Loggen skall vara en vanlig sekvensfil med fast postlängd av 80 tecken. Du kan allokera minimalt med utrymme.
- Namnet på loggen skall vara `userid.KURS.ARTLOG`
- När du läst samtliga poster i artikelregistret så skall innehållet i loggen skrivas ut till Sysout.
- Loggen skall ha följande utseende

Logg för artiklar, utskriven dd/mm, 20xx

Artnr	Beskrivning	Feltyp
40910	CIG. TÄNDARE, ORDINÄR	** Ejnum **
73081	PILSNER	** Ejnum **

Slut på logg

- Kompilera programmet.
- Exekvera programmet och kontrollera resultatet.

Uppgiften är avslutad.

Övningen är avslutad.

Övning 22 - Filmatchning

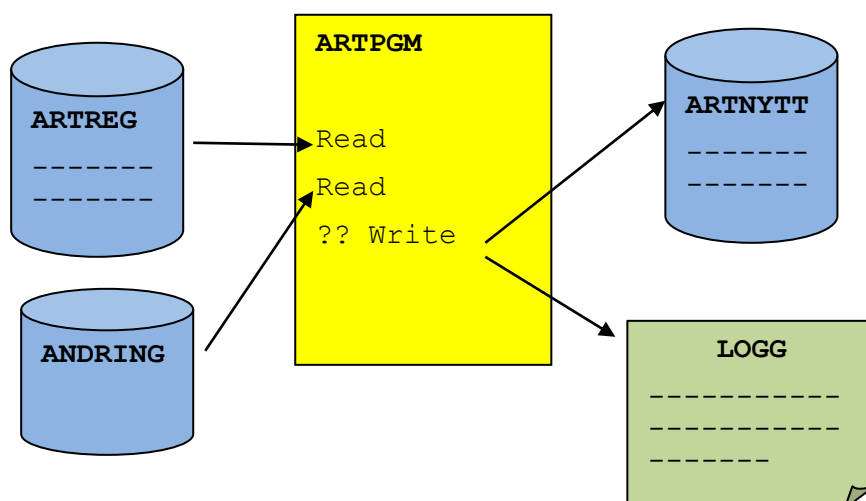
I denna övning kommer du att skriva ett nytt program som läser in ändringstransaktioner för artikelregistret från en speciell ändringslogg och skapar ett nytt, uppdaterat artikelregister.

Uppgift 1 - Ändringstransaktioner

- I filen med namnet `instruktör.KURS.ARTIKLAR.ANDRING` så finns det olika transaktioner som avser förändringar i artikelregistret.
- Posternas utseende framgår av denna bild:

```
U 10010 DYRK, VOLVO          A 13000004140. . . . .  
A 21120 BATTERI, MERCEDES    A 1700000 . . . . .  
R 32005
```

- Innebörden av de olika koderna är följande:
U = Uppdatera
A = Addera
R = Radera
- Du skall läsa ändringar från ändringsfilen och matcha dessa mot artikelregistret, som du tidigare arbetat med.
- När du läser en post från det gamla registret skall du kontrollera om det finns en ändring som avser detta artikelnummer. I så fall skall du vidta åtgärden som transaktionskoden indikerar. Om det inte är någon ändring, så skriver du den gamla posten till det nya registret.
- Ändringarna är sorterade i artikelnummer.



- Skapa ett nytt artikelregister under TSO/ISPF med namnet `userid.KURS.ARTNYTT`. Du kan välja samma filattribut som det gamla artikelregistret.

- För varje ändring så skall du skriva i ändringsloggen, som du använde i föregående uppgift.
- När du är klar så skall du skriva ut loggen till sysout.

Logg för artiklar, utskriven dd/mm, 20xx		
Artnr	Beskrivning	Feltyp
10010	DYRK, VOLVO	** Uppdat **
21120	BATTERI, MERCEDES	** Adderad**
40910	CIG. TÄNDARE, ORDINÄR	** Ejnum **
32005	KATTFOT	** Raderad**
73081	PILSNER	** Ejnum **
Slut på logg		

Uppgiften är avslutad.

Övningen är avslutad.

Övning 23 - Lagringsformer

I denna övning kommer du att skriva ett nytt program som skriver några poster med olika numeriska dataformat till en fil. Denna fil skall Du sedan editera och undersöka.

Uppgift 1 – Skapa fil

- Skapa en ny fil med ISPF. Filens namn skall vara *userid.AP15.PICUT*.
- Postlängd skall vara 80, bestäm själv storleken på filen (minimal).
- Verifiera att filen har skapats med korrekta filattribut.

Uppgiften är avslutad.

Uppgift 2 – Skriv program

- Du skall skapa ett program med namnet OVN231 i ditt vanliga COBOL-bibliotek
- Programmet skall kunna skriva poster till filen som Du skapade i Uppgift 1.
- Skriva en rad med ledtext samt data enligt följande:

```
Detta är en textsträng i formatet PIC X(08) Value 'COBOL' *** COBOL ***
```

- Skriva ytterligare några rader med olika numeriska format såsom Zonat Decimalt, Packat Decimalt samt Binärt, med både positiva och negativa värden. Se exempel nedan.

```
Pic 9(04) Value 1234          *** 1234 ***
```

- Kompilera och exekvera ditt program.
- Editera filen *userid.AP15.PICUT* och försätt editorn i hexadecimalt läge :
HEX ON
- Studera de olika posternas utseende med avseende på de numeriska värdena.

Uppgiften är avslutad.

Övningen är avslutad.