

UMEÅ UNIVERSITET  
Institutionen för datavetenskap  
Thomas Johansson



November 1997

# Introduktion till Borland C/C++ 4.5 Version 2

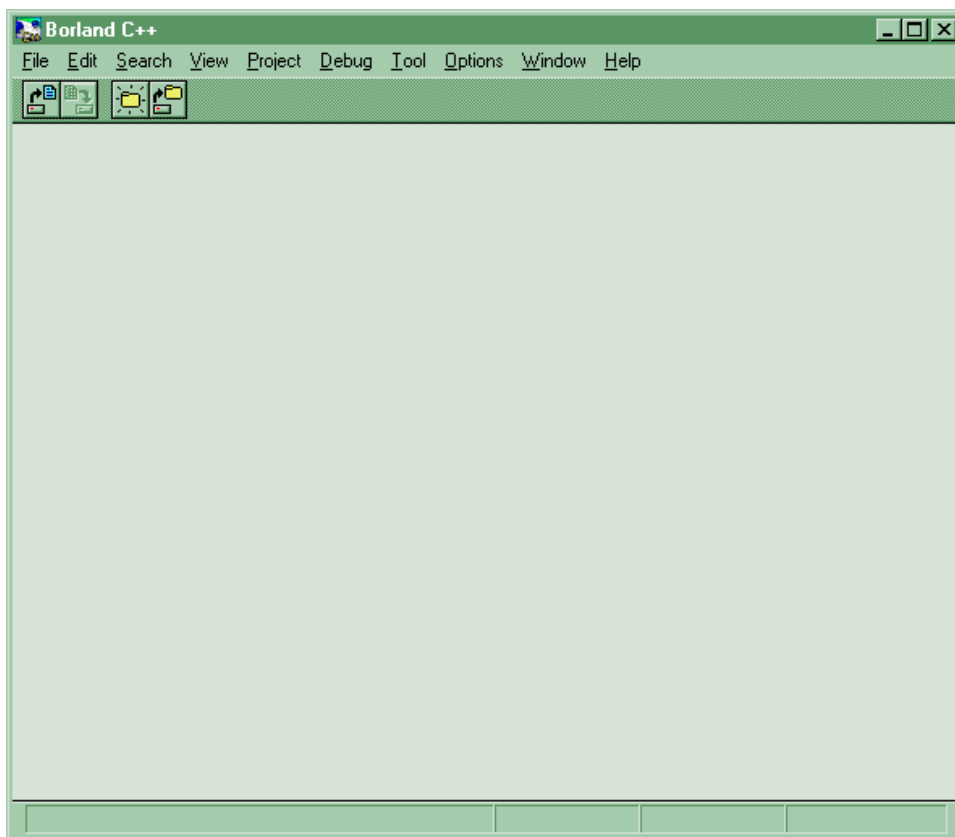
Inledning.....	3
Starta Borland C .....	3
Nytt projekt .....	5
Kompilera .....	8
Körningsresultat .....	10
Stoppa programmet.....	10
Avsluta.....	10
Debugger.....	11

## Inledning

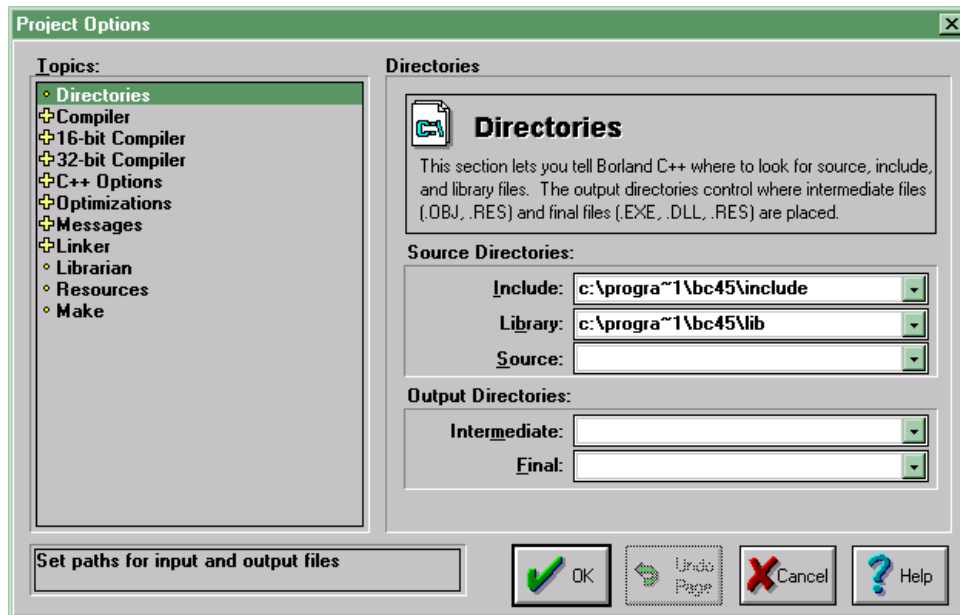
Detta häfte beskriver översiktligt hur du kommer igång med Borland C/C++ version 4.5. Beskrivningen gäller för både vanliga Windows 3.11 ('Workgroups') och för Windows 95/NT. Bilderna är hämtade från Windows NT och kan se något annorlunda ut under 3.11.

## Starta Borland C

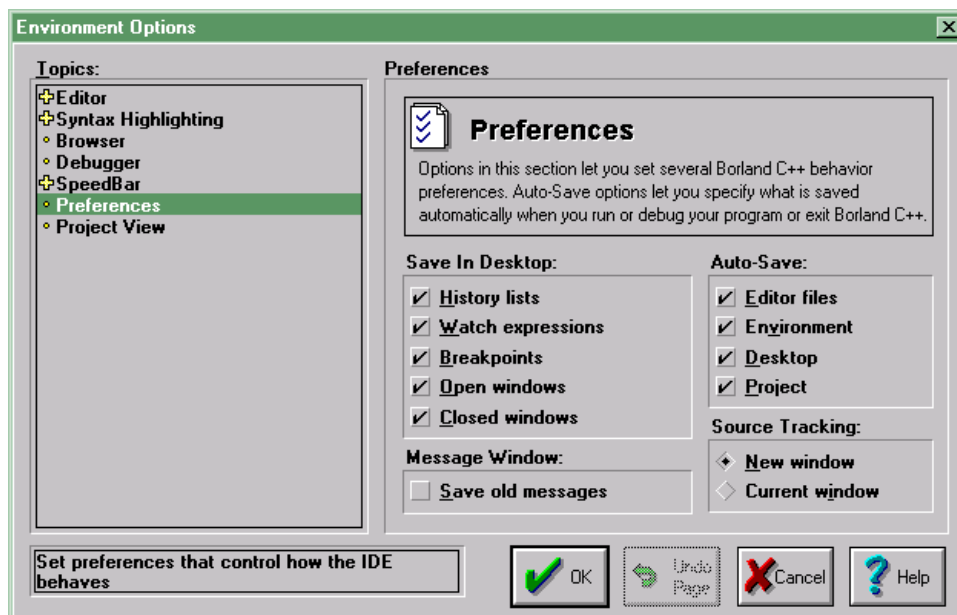
Starta programmet genom att dubbelklicka på ikonen . Efter en stund visas följande fönster:



Beroende på hur den förre användaren avslutade kanske det finns flera fönster öppna inuti det stora fönstret, stäng i så fall dessa. Innan du börjar med att arbeta med Borland C bör du kontrollera en del inställningar. Det är troligt att de redan är rätt inställda, men det är bäst att vara på den säkra sidan. På menyraden finns ett menyval som heter **Options**, välj **Environment** under den meny. Du ser då ett fönster med en lista val under **Topics**:



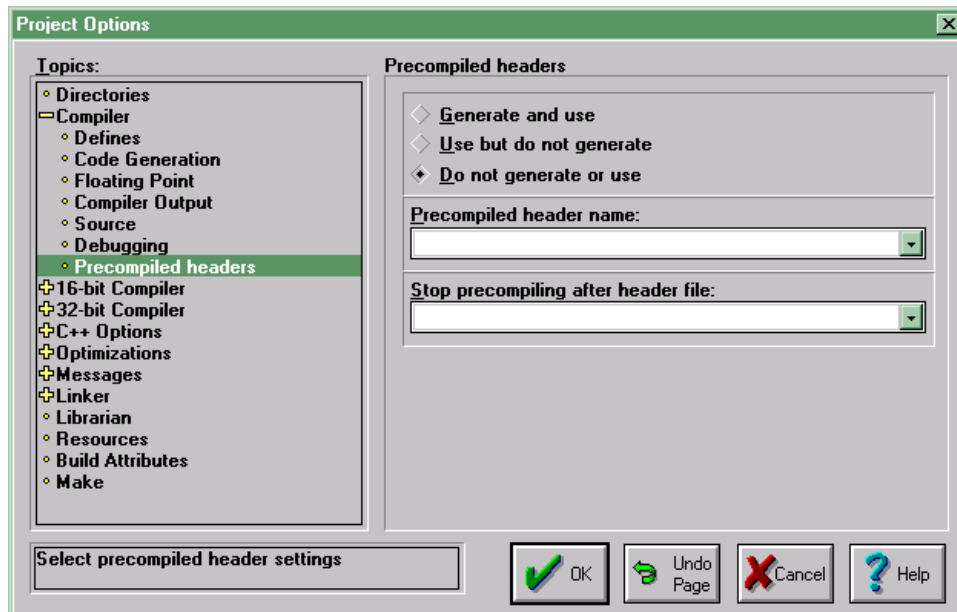
Markera **Preferences** under **Topics** och kontrollera att rutorna är ikryssade som i exemplet:



När du blir mer van vid utvecklingsmiljön kan du ändra på dessa inställningar, men låt dem vara så här tills vidare. Den viktigaste rutan är **Editor files** under rubriken **Auto-Save**, med den ikryssad sparas ditt program automatiskt innan du kompilerar och kör det. Det kommer säkert att hända att datorn 'hänger upp sig' när du provar ditt program, och om du måste starta om den då är det bra att dina senaste ändringar sparades innan.

En annan inställning som kan vara bra att göra om du tänker spara projektet på diskett är att ta bort 'precompiled headers'. Dessa tar en hel del plats och kan

lätt orsaka utrymmesbrist på disketten. Välj **Compiler** under **Topics** och sedan **Precompiled Headers** :

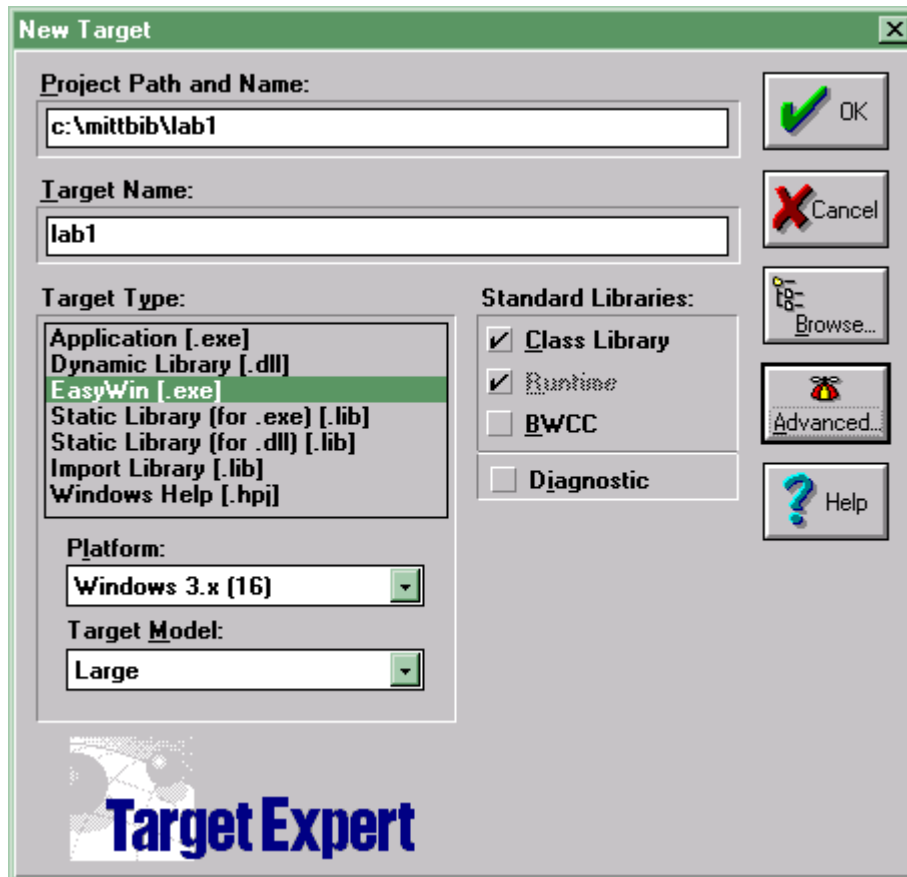


Genom att klicka för 'Do not generate or use' så förhindrar du att Borland C använder precompiled headers.

Klicka på **OK** för att spara inställningarna.

## Nytt projekt

Nu är du klar för att börja med ditt program. I Borland C finns något som kallas **projekt** som 'håller ihop' de olika delarna i ett program. Speciellt när dina program växer och består av flera filer är det mycket praktiskt att systemet håller reda på dem åt dig. Under menyn **Project** väljer du **New Project**. Du ser då följande fönster (i detta exempel är projektets namn redan ifyllt):

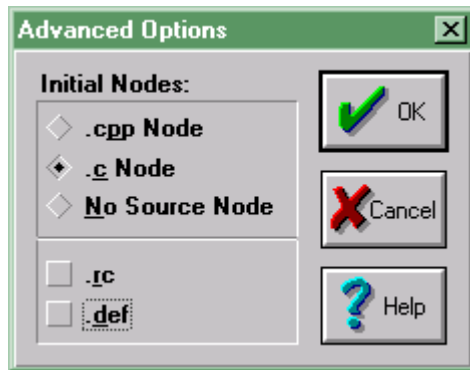


Här väljer du vad ditt program skall heta, var i din dator det skall ligga och vilken sorts projekt det är. Under rubriken **Project Path and Name** skriver du in sökvägen till den katalog där du vill att projektet skall ligga, och namnet på ditt projekt. Det är lämpligt att använda en ny katalog för varje program, exempelvis en katalog för laboration1, en ny för lab 2 osv. Katalogen behöver inte finnas innan du skapar projektet, det ordnar systemet. Observera att det är viktigt att du väljer ett ställe som du har skrivrättigheter till, exempelvis hårddisken på datorn (C: ), en diskett (A: ) eller din nätverksenhet om du har någon sådan uppkopplad. Obs att vissa datorer i labbsalarna inte tillåter lagring av filer på C:. Att använda diskett går bra men medför att det tar längre tid att spara och hämta program och även att kompilera dem.

**Target Name** blir automatiskt samma som **Project Name**, det kan du ändra men det är praktiskt att ha samma här.

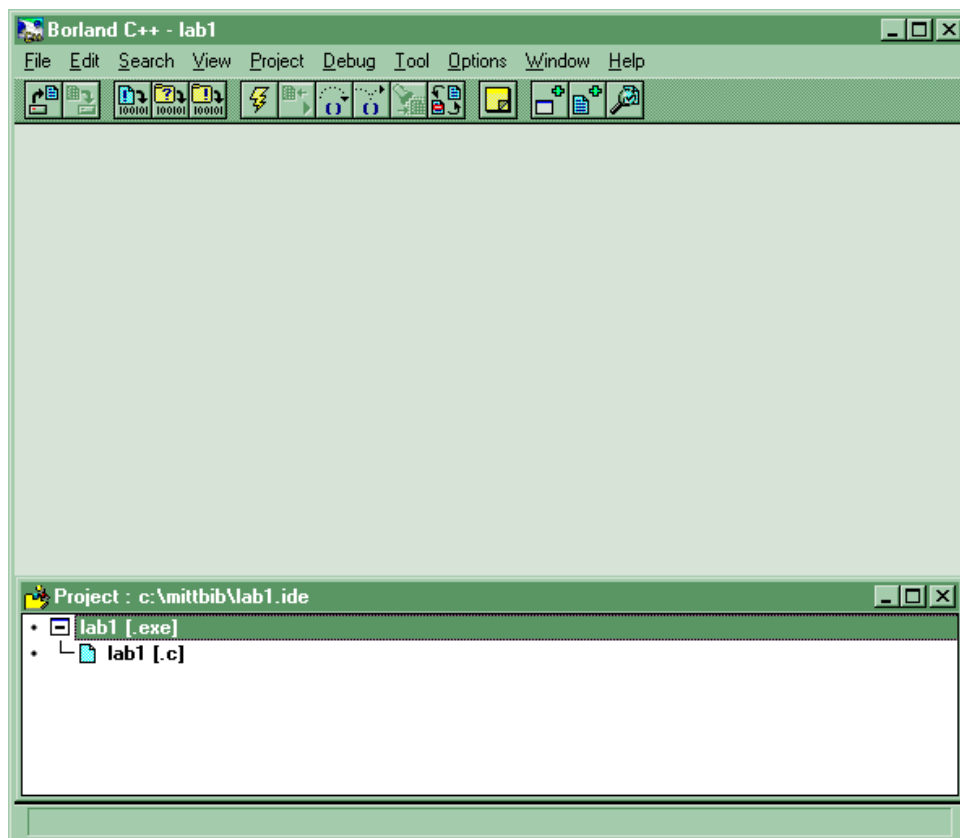
**Target Type** skall vara **EasyWin**, det finns flera andra sorters programtyper du kan göra men det absolut enklaste är att välja denna typ.

Slutligen, klicka på **Advanced ...** och du får upp denna bild:

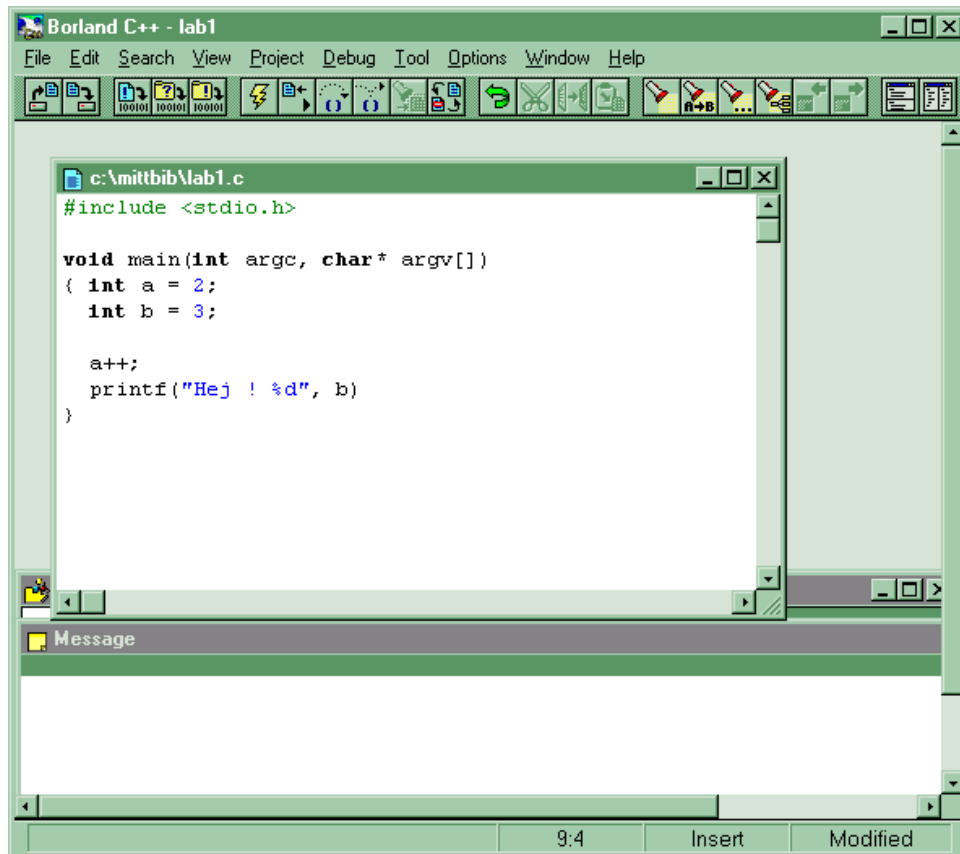


Om du skall skriva ett vanligt c-program är detta lämpliga inställningar, ändra så att det ser ut som på bilden. Markera .cpp om du skriver i C++.

Tryck **OK** här och sedan **OK** i Target Expert-fönstret och du har ett nytt projekt:



Systemet har gjort ett projekt med en **nod** (lab1.exe) som skapas från en **c-fil** (lab1.c). I större program kommer det att finnas fler rader i ditt projekt. Dubbelklicka på raden lab1.c för att börja skriva in ditt program:



Nästa gång du vill arbeta med projekt lab1 väljer du istället **Open project** under **Project**-menyn, då kommer du tillbaka hit och din skärm ser ut så som den såg ut när du avslutade Borland C förra gången. Projektet innehåller inte bara information om vilka filer som hör till, utan också om de inställningar du gjort.

## Kompilera

Efter en stund kanske du vill prova att kompilera ditt program för att se om du har gjort rätt hittills. Det är inte alls nödvändigt att skriva färdigt hela programmet innan du prov-kompilerar, men många fel som kompilatorn upptäcker orsakar i sin tur fler fel längre ned i programmet, det är vanligt med 25 följdfelet eller mer.

Det finns flera sätt att starta kompileringen av programmet:

Välj **Make** i **Project**-menyn

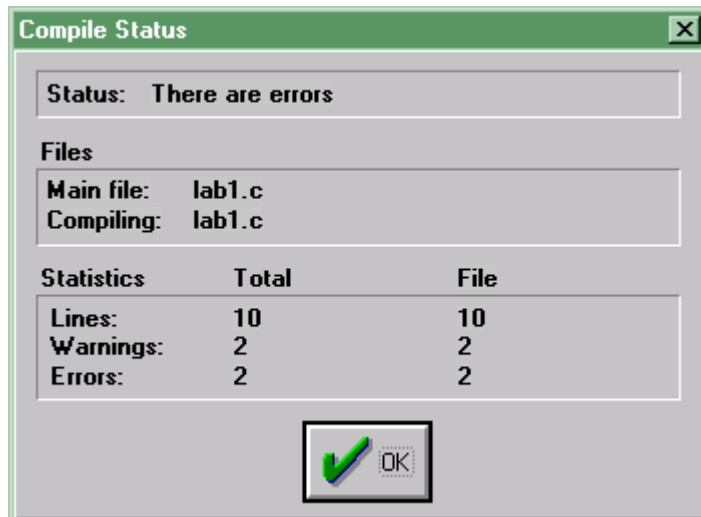
Välj **Run** i **Debug**-menyn eller tryck **Ctrl-F9**

Tryck på knappen med den gula blixten (samma som **Run**)

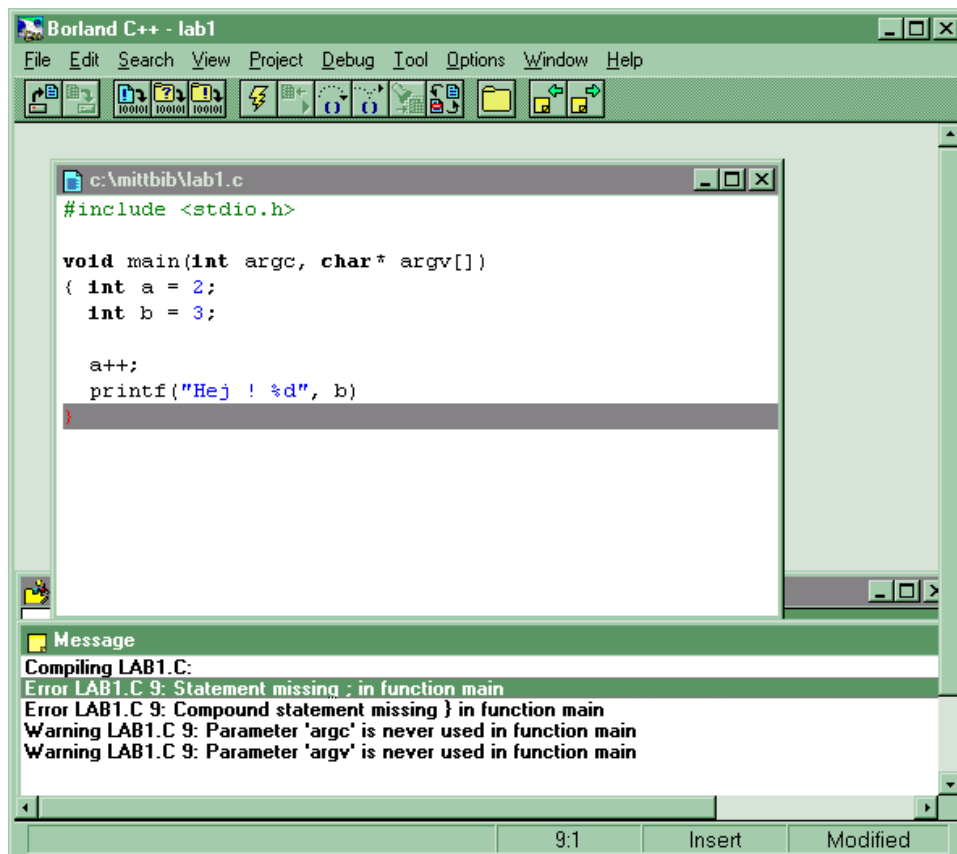
Om du väljer run kommer programmet att startas om det klarade sig igenom kompileringen utan fel.

Efter kompilering kommer en lista med fel och varningar upp:





Tryck **OK** för att se vad som gick fel: (Rutan kommer även om kompileringen gick bra.)



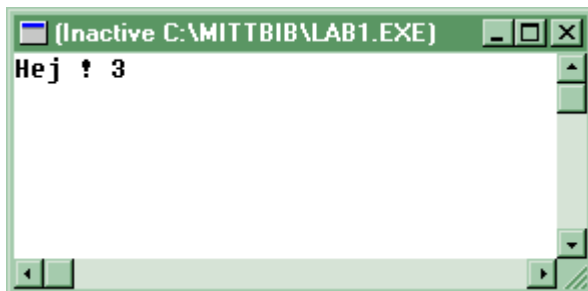
Genom att klicka på felen i **Message**-fönstret hoppar du automatiskt till det ställe i programfilen där kompilatorn upptäckte att något inte var som det skulle. Det är dock inte helt säkert att det är just där felet verkligen finns, det kan vara något på en rad tidigare i programmet som är fel, ett felstavat variabelnamn i en deklARATION till exempel. I detta fall saknas ett semikolon men det leder till att kompilatorn inte hittar den avslutande 'krullparentesen' (}).

Varningar hindrar inte programmet från att köras, men det är en god vana att alltid titta efter varför de finns. I detta exempel varnas för att två parametrar inte används, vilket är som det skall just här.

"Linker Warning: No module definition ..." är en varning som alltid dyker upp vid denna typ av applikation (EasyWin), när kompileringen gått bra. Länkaren hittar ingen 'definitionsfil' utan använder standardvärden för bland annat stackstorlek i stället, vilket fungerar bra i de flesta fall.

## Körningsresultat

För att kunna få med utskrifter från program i laborationsrapporter etc är det enklast att klippa ut dem och sedan klistra in dem i ordbehandlingsprogrammet. Det är inte möjligt att bara klippa ut texten ur ett fönster, däremot går det bra att klippa ut hela fönstret. Se till att fönstret är aktivt (klicka på titelraden) och tryck sedan på **Alt + PrintScreen**. Innehållet i fönstret, plus ram och titelrad, finns nu i urklippet. Byt till ordbehandlaren och klistra in med **Ctrl + v**. Dokumentet kan bli ganska stort på detta vis, så klistra inte in onödigt stora fönster. Denna introduktion t.ex har nu blivit över 3 MByte på grund av alla bilder och ryms därför inte på en diskett.

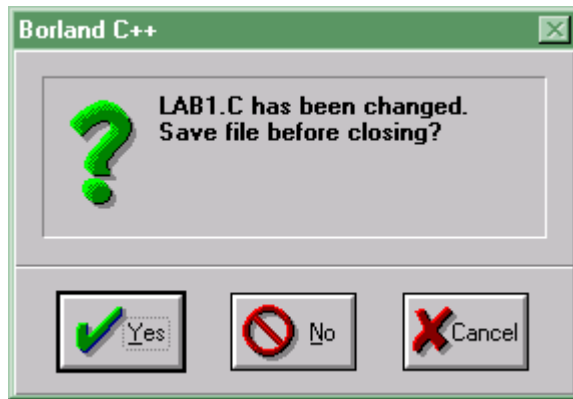


## Stoppa programmet

När programmet har kört färdigt står texten (Inactive ...) i titelraden, då kan fönstret stängas. Om programmet har råkat in i en oändlig loop så prova med **Ctrl + c**, **Ctrl + Break**, **Terminate program** under **Debug-menyn**, eller att stänga utmatningsfönstret. Fungerar inte detta så återstår **Ctrl + Alt + Delete**. Beroende på operativsystem (Windows 3.11, -95 eller NT) så uppför sig datorn olika, men det finns en god chans att slippa starta om allting.

## Avsluta

När du till slut är nöjd med ditt arbete så avsluta Borland C genom att välja **Exit** i **File**-menyn. Om det då finns några ändringar som inte är sparade får du denna fråga:



Klicka på **YES** så sparas ditt arbete. Om du ändrar dig och inte vill avsluta så klicka på **Cancel**.

## Debugger

Debuggern ('avlusaren') i Borland C är ett bra hjälpmedel för att hitta logiska fel i program. I detta avsnitt förklaras de grundläggande kommandon som kan ges till debuggern.

Under menyn **Debug** finns bland annat följande val:

- Run                      Ctrl-F9
- Step over                F8
- Trace into               F7
- Toggle breakpoint     F5
- Add watch               Ctrl-F5
- Add breakpoint
- Inspect

### Run

Kompilerar (om nödvändigt) programmet och kör det sedan.

### Step over

Kör aktuell rad och stegar sedan fram till nästa rad i programmet och stannar där. Om aktuell rad är ett funktionsanrop så körs funktionen i sin helhet.

### Trace into

Samma som *Step over* utom i de fall aktuell rad är ett funktionsanrop, då stegas aktuell rad fram till första raden i den anropade funktionen.

### Toggle breakpoint

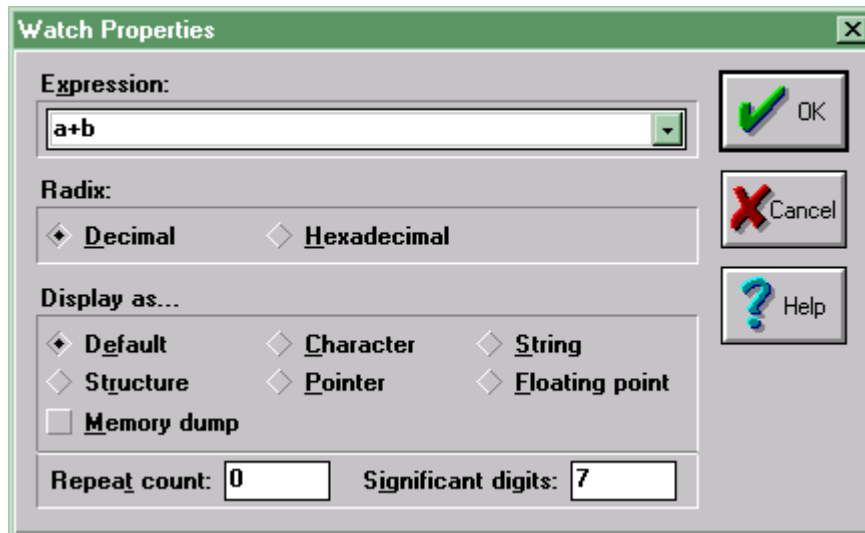
En breakpoint ('brytpunkt') är en markering som kan sättas på en rad. När datorn sedan kommer fram till denna rad efter ett run-kommando så stannar den där. *Toggle breakpoint* slår av och på en brytpunkt på aktuell rad.

## Add breakpoint

Sätter en brytpunkt på aktuell rad, ungefär som *Toggle breakpoint*

## Add watch

Du kan lägga in ett 'övervakningsfönster' som visar ett uttrycks värde under programkörningen. Välj **Help** om du vill veta mer om *watch*.



## Inspect

Liknar *watch* men 'scope' för de undersökta variablerna är annorlunda; titta i hjälpen för att se skillnaden.

