File Processing

Introduction

- Programs shown up to this point had involved relatively small amount of input/output data
- Input data were read from a terminal and output was displayed in the terminal
- This is adequate if the volume of the data involved is not large

Introduction Contd..

- Applications involving large data sets can be processed more conveniently if the data is stored in files

File Basics

- Any collection of data items that is input to or output by a program is called a file

- Each line of data in a file is called a record

Opening Files

- Before a file can be used for input or output in a Fortran program – it must be opened

- This can be accomplished using an open statement of the form

  open*(open-list)*

open-list

- Includes
  - A unit specifier
  - A file = clause
  - A status = clause
  - An action=clause

Unit Specifier

- Has the form:

  unit=*integer-expression*

  or simply

  integer-expression

- The value of integer-expression is a nonnegative number that designates the unit number to be connected to this file
- Reference to this file by read or write statements is by means of this unit number

file = clause

- Has the form:

  file = *character-expression*

- The value of the character-expression is the name of the file

status = clause

- Has the form:

  status = *character-expression*

- The value of the *character-expression* must be one of the following:

  old

  new

  replace

  scratch

action = clause

- Has the form:

  action = *character-expression*

- The value of the *character-expression* must be one of the following:

  read

  write

  readwrite

Illustration of open statement

```
open(unit=10,file =’info1’,status=’old’)

character(10) :: infile
print*,’Enter name of input file’
read*, infile
open (unit=11,file=infile,status=’old’)

open(unit=12,file =’info2’,status=’new’)
open(unit=13,status=’scratch’)
```

Closing Files

- The close statement is used to disconnect a file from its unit number

- This statement is of the form:

  close(*close-list*)

close-list

- Must include:
  - A unit specifier
- May include:
  - An iostat = clause
  - An err = clause
  - A status = clause

```
close(10)
```

File Input

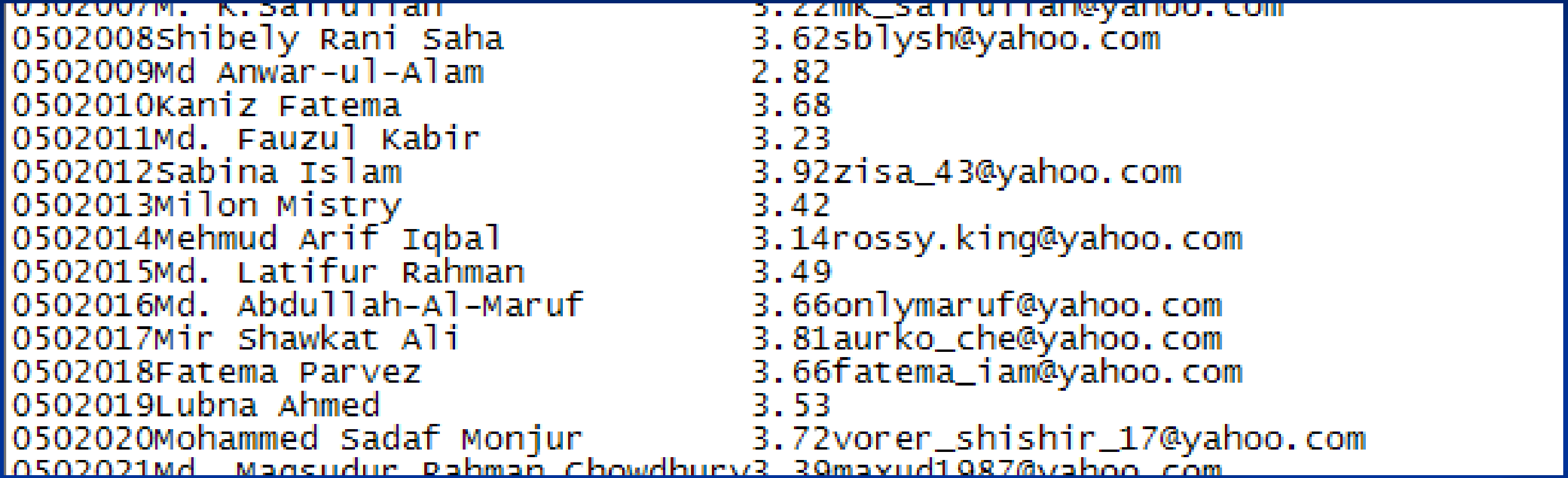
- Data can be read from a file using a READ statement of the general form:

  read (*control-list*) *input-list*

control-list

- Must include
  - A unit specifier indicating the unit number connected to the file
- May include
  - A format specifier
  - An iostat = clause
  - An end = clause

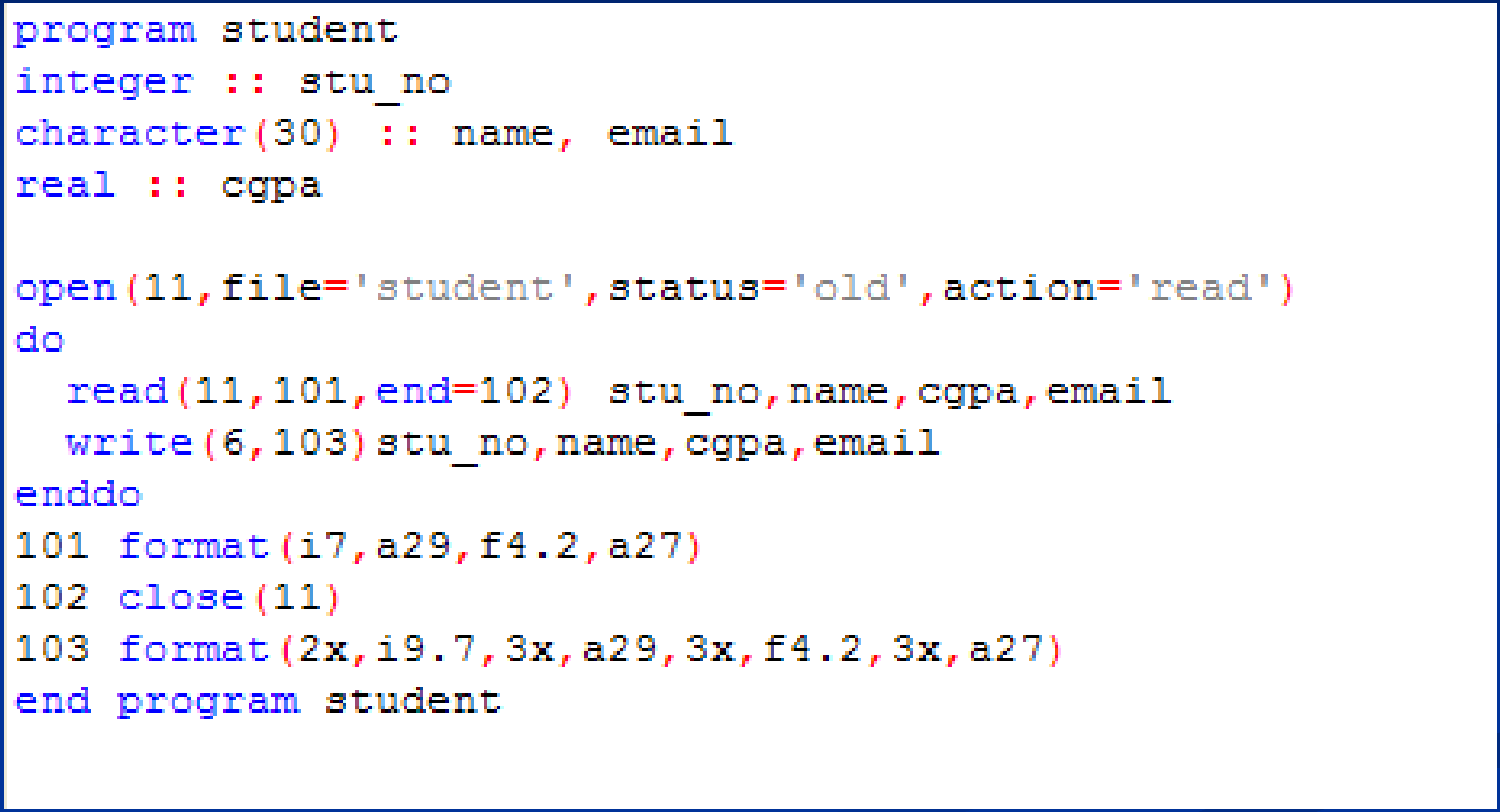
Sample File(formatted) - student



```
0502007M. K.Saifullah               3.22mk_saifullah@yahoo.com
0502008Shibely Rani Saha            3.62sblysh@yahoo.com
0502009Md Anwar-ul-Alam             2.82
0502010Kaniz Fatema                 3.68
0502011Md. Fauzul Kabir             3.23
0502012Sabina Islam                 3.92zisa_43@yahoo.com
0502013Milon Mistry                 3.42
0502014Mehmud Arif Iqbal            3.14rossy.king@yahoo.com
0502015Md. Latifur Rahman           3.49
0502016Md. Abdullah-Al-Maruf        3.66onlymaruf@yahoo.com
0502017Mir Shawkat Ali              3.81aurko_che@yahoo.com
0502018Fatema Parvez                3.66fatema_iam@yahoo.com
0502019Lubna Ahmed                  3.53
0502020Mohammed Sadaf Monjur        3.72vorer_shishir_17@yahoo.com
0502021Md. Maqsudur Rahman Chowdhury3.39maxud1987@yahoo.com
```

1-7	Student No.
8-36	Name
37-40	CGPA
41-67	email

Program



```
program student
integer :: stu_no
character(30) :: name, email
real :: cgpa

open(11,file='student',status='old',action='read')
do
  read(11,101,end=102) stu_no,name,cgpa,email
  write(6,103)stu_no,name,cgpa,email
enddo
101 format(i7,a29,f4.2,a27)
102 close(11)
103 format(2x,i9.7,3x,a29,3x,f4.2,3x,a27)
end program student
```

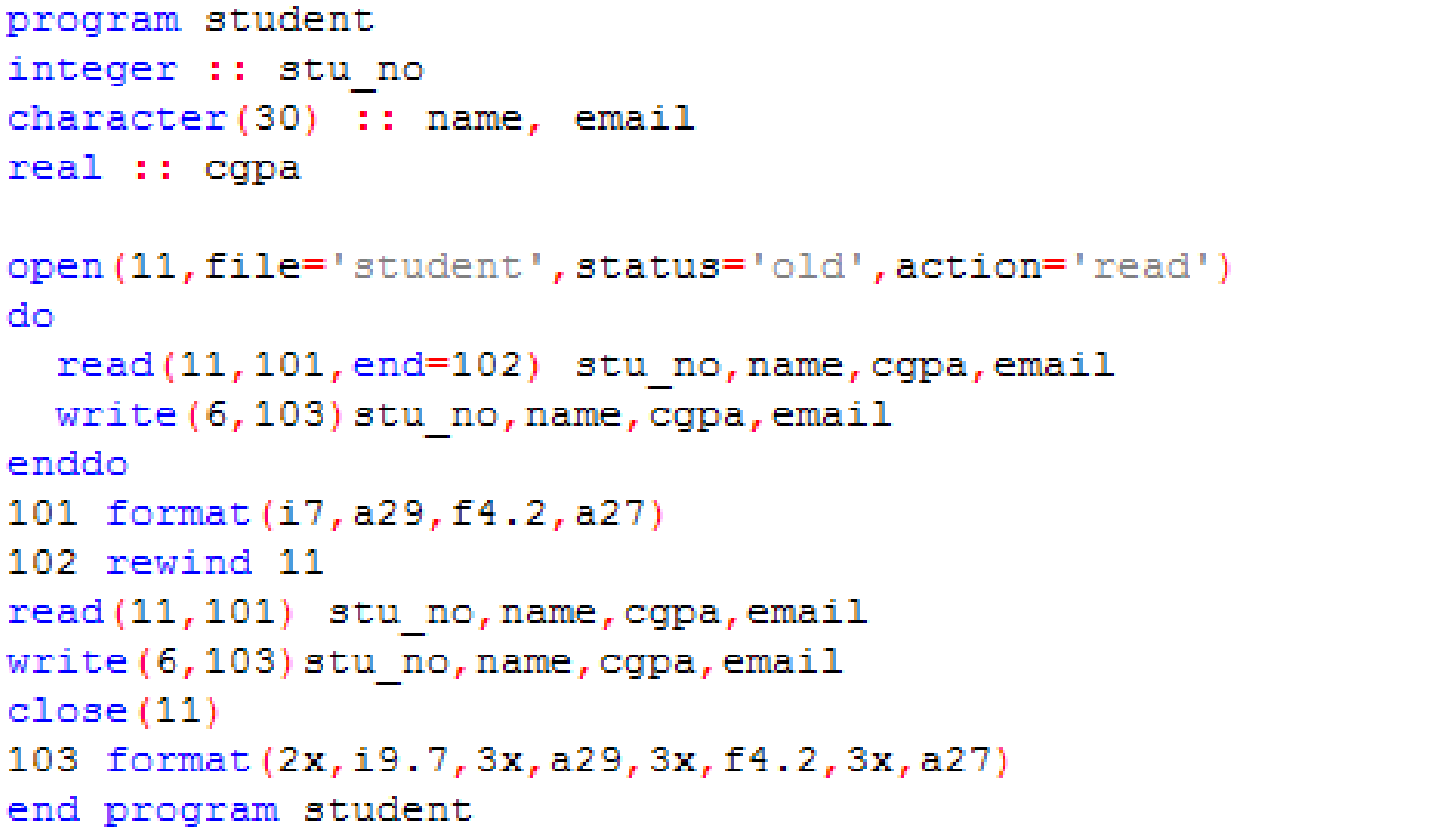
File-positioning Statements

- rewind *unit* or rewind(*position-list*)
- backspace *unit* or backspace(*position-list*)
- endfile *unit* or endfile(*position-list*)

position-list

- Must contain
  - A unit specifier
- May contain
  - An iostat = clause

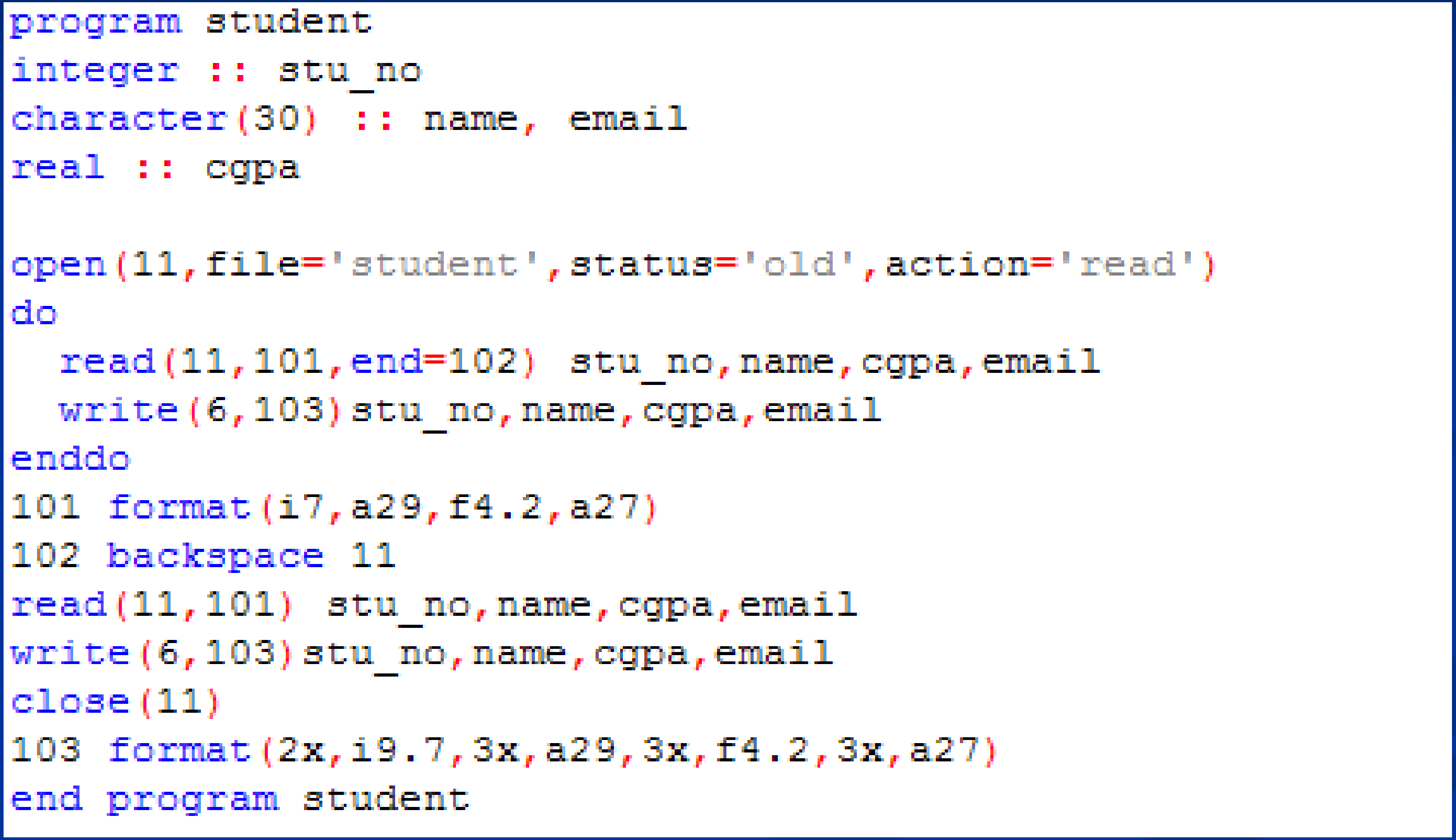
Modified Program with rewind



```
program student
integer :: stu_no
character(30) :: name, email
real :: cgpa

open(11,file='student',status='old',action='read')
do
  read(11,101,end=102) stu_no,name,cgpa,email
  write(6,103)stu_no,name,cgpa,email
enddo
101 format(i7,a29,f4.2,a27)
102 rewind 11
read(11,101) stu_no,name,cgpa,email
write(6,103)stu_no,name,cgpa,email
close(11)
103 format(2x,i9.7,3x,a29,3x,f4.2,3x,a27)
end program student
```

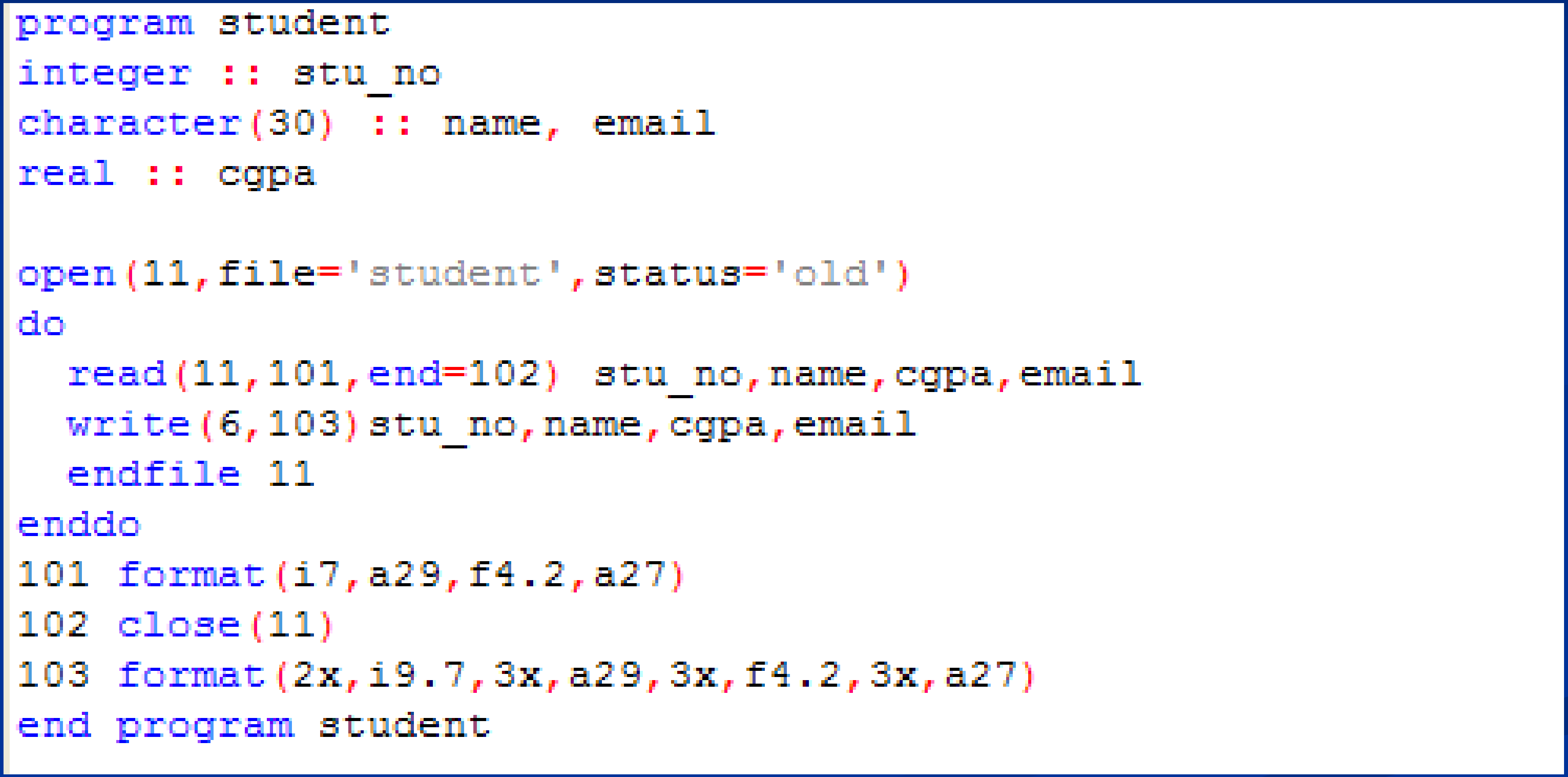
Modified Program with backspace



```
program student
integer :: stu_no
character(30) :: name, email
real :: cgpa

open(11,file='student',status='old',action='read')
do
  read(11,101,end=102) stu_no,name,cgpa,email
  write(6,103)stu_no,name,cgpa,email
enddo
101 format(i7,a29,f4.2,a27)
102 backspace 11
read(11,101) stu_no,name,cgpa,email
write(6,103)stu_no,name,cgpa,email
close(11)
103 format(2x,i9.7,3x,a29,3x,f4.2,3x,a27)
end program student
```

Modified Program with endfile



```
program student
integer :: stu_no
character(30) :: name, email
real :: cgpa

open(11,file='student',status='old')
do
   read(11,101,end=102) stu_no,name,cgpa,email
   write(6,103)stu_no,name,cgpa,email
   endfile 11
enddo
101 format(i7,a29,f4.2,a27)
102 close(11)
103 format(2x,i9.7,3x,a29,3x,f4.2,3x,a27)
end program student
```

File Output

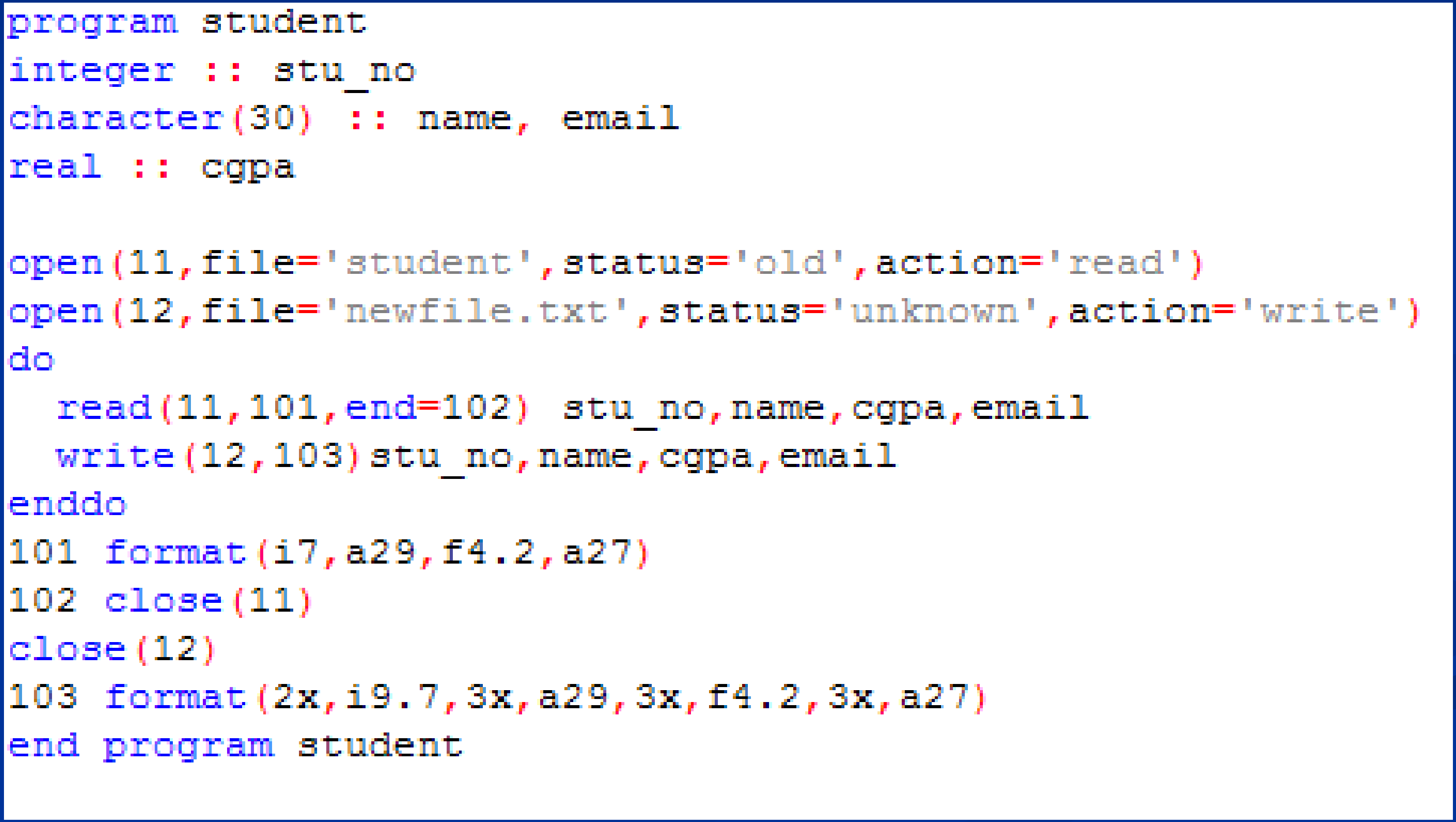
- Data is written to a file using a write statement of the form

  write(*control-list*) *output-list*

control-list

- Must include
  - A unit specifier indicating the unit number connected to the file
- May include
  - A format specifier
  - An iostat = clause

File Output Example



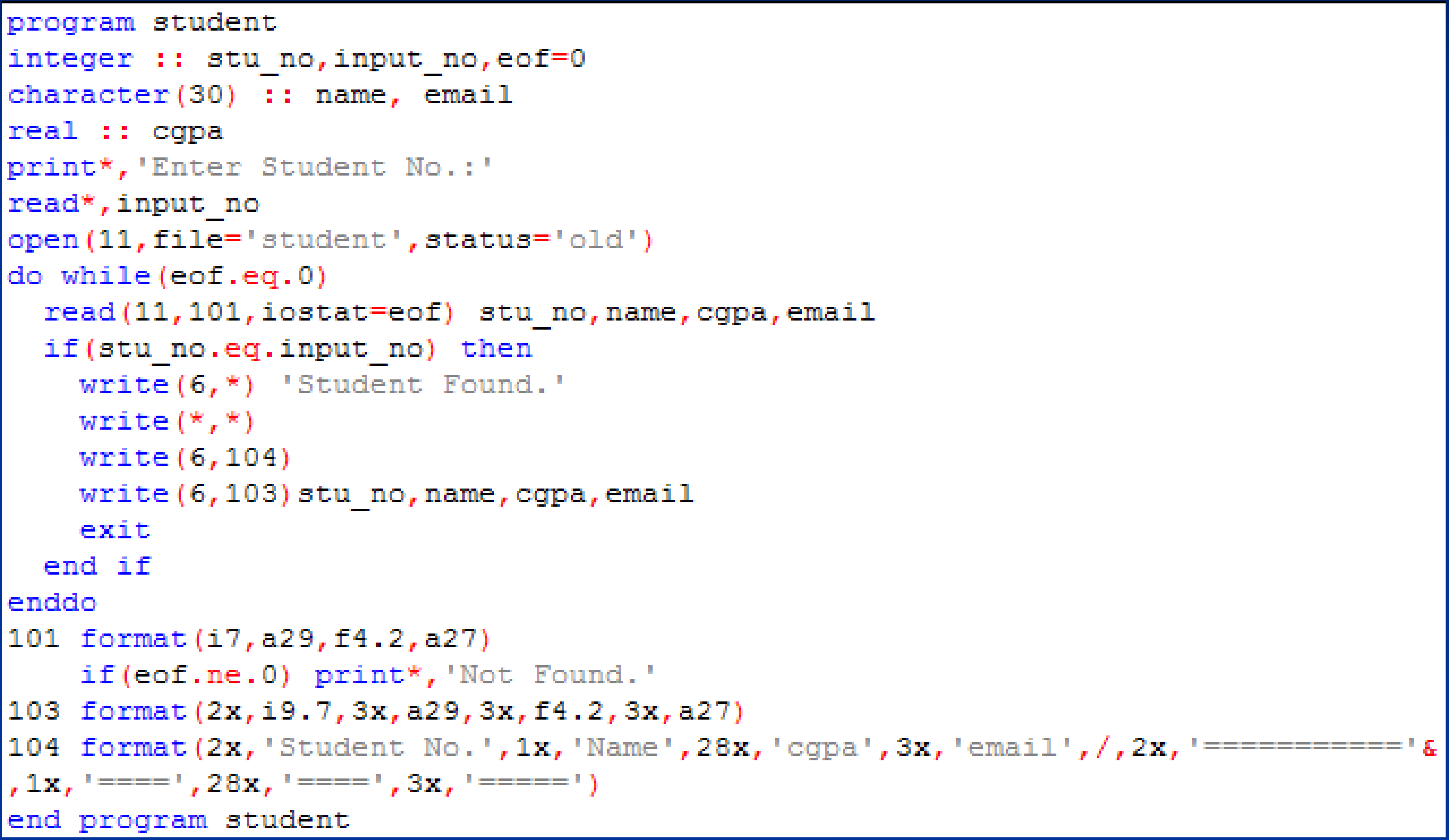
```
program student
integer :: stu_no
character(30) :: name, email
real :: cgpa

open(11,file='student',status='old',action='read')
open(12,file='newfile.txt',status='unknown',action='write')
do
  read(11,101,end=102) stu_no,name,cgpa,email
  write(12,103)stu_no,name,cgpa,email
enddo
101 format(i7,a29,f4.2,a27)
102 close(11)
close(12)
103 format(2x,i9.7,3x,a29,3x,f4.2,3x,a27)
end program student
```

Searching

- A program that reads student no. and then prints all available information about the student

Program



```
program student
integer :: stu_no,input_no,eof=0
character(30) :: name, email
real :: cgpa
print*,'Enter Student No.:'
read*,input_no
open(11,file='student',status='old')
do while(eof.eq.0)
  read(11,101,iostat=eof) stu_no,name,cgpa,email
  if(stu_no.eq.input_no) then
    write(6,*) 'Student Found.'
    write(*,*)
    write(6,104)
    write(6,103)stu_no,name,cgpa,email
    exit
  end if
enddo
101 format(i7,a29,f4.2,a27)
    if(eof.ne.0) print*,'Not Found.'
103 format(2x,i9.7,3x,a29,3x,f4.2,3x,a27)
104 format(2x,'Student No.',1x,'Name',28x,'cgpa',3x,'email',/,2x,'============'&
,1x,'====',28x,'====',3x,'=====')
end program student
```