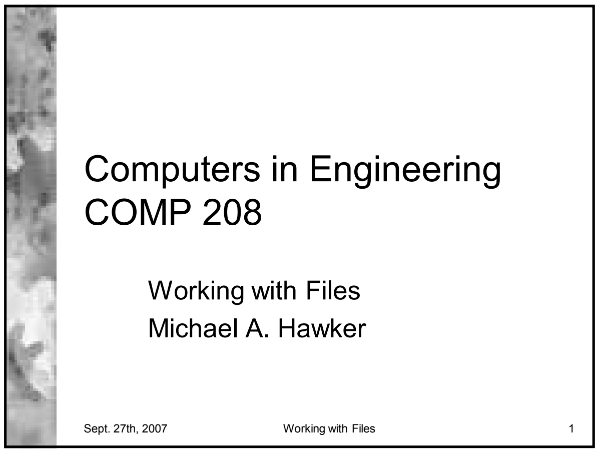
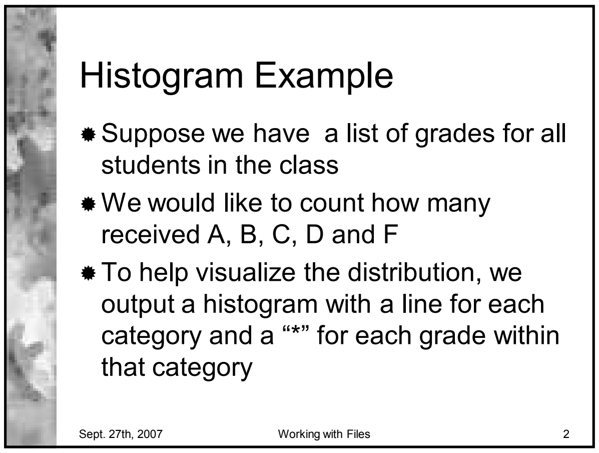


Computers in Engineering COMP 208

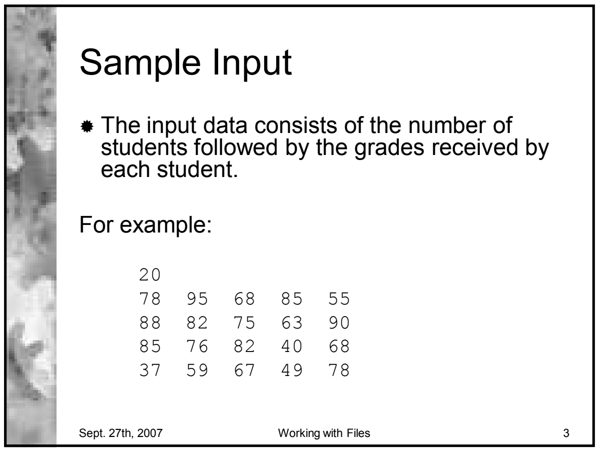
Working with Files

Michael A. Hawker



Histogram Example

- Suppose we have a list of grades for all students in the class
- We would like to count how many received A, B, C, D and F
- To help visualize the distribution, we output a histogram with a line for each category and a "*" for each grade within that category

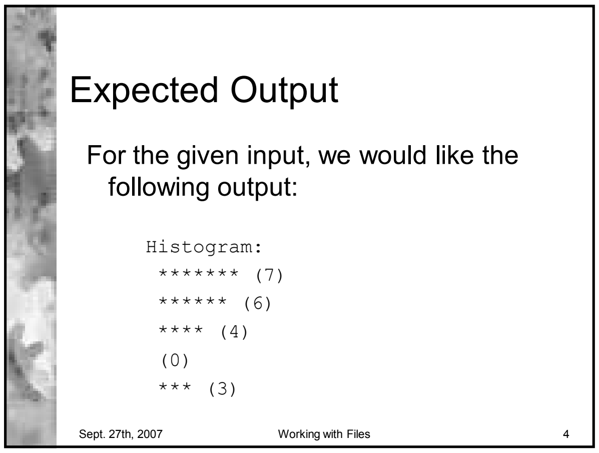


Sample Input

- The input data consists of the number of students followed by the grades received by each student.

For example:

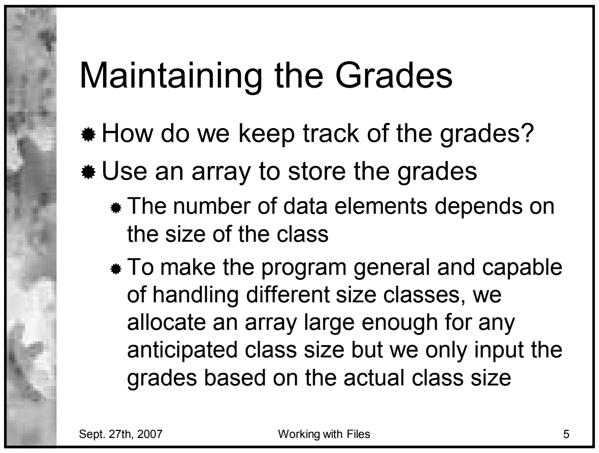
```
20
78  95  68  85  55
88  82  75  63  90
85  76  82  40  68
37  59  67  49  78
```



Expected Output

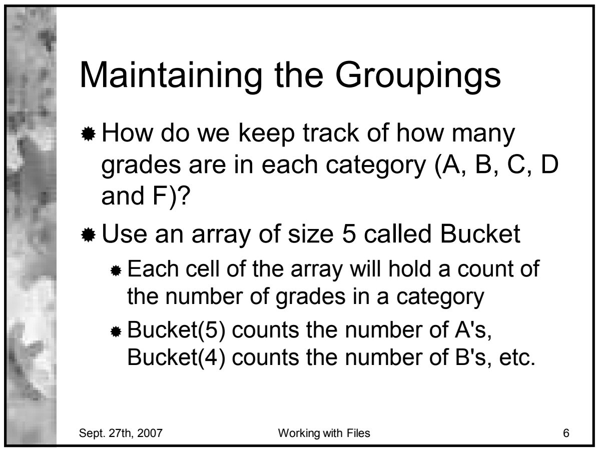
For the given input, we would like the following output:

```
Histogram:
 ******* (7)
 ****** (6)
 **** (4)
 (0)
 *** (3)
```



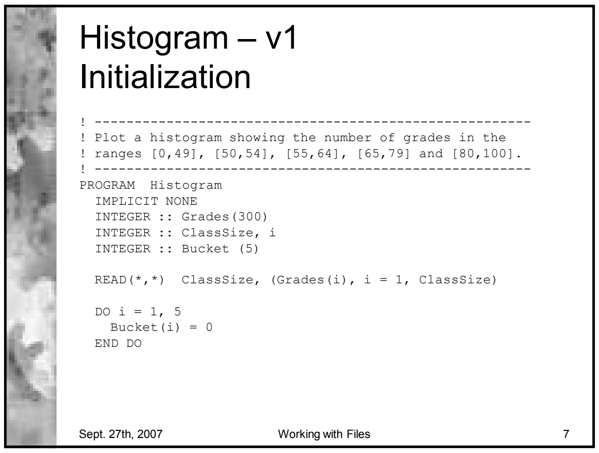
Maintaining the Grades

- How do we keep track of the grades?
- Use an array to store the grades
  - The number of data elements depends on the size of the class
  - To make the program general and capable of handling different size classes, we allocate an array large enough for any anticipated class size but we only input the grades based on the actual class size



Maintaining the Groupings

- How do we keep track of how many grades are in each category (A, B, C, D and F)?
- Use an array of size 5 called Bucket
  - Each cell of the array will hold a count of the number of grades in a category
  - Bucket(5) counts the number of A's, Bucket(4) counts the number of B's, etc.



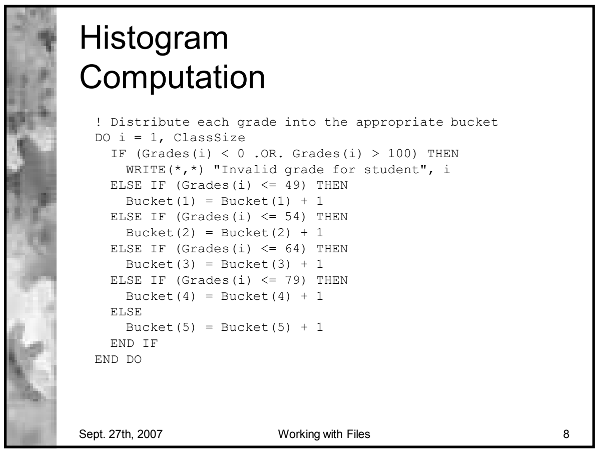
Histogram – v1 Initialization

```
! -------------------------------------------------------
! Plot a histogram showing the number of grades in the
! ranges [0,49], [50,54], [55,64], [65,79] and [80,100].
! -------------------------------------------------------
PROGRAM  Histogram
  IMPLICIT NONE
  INTEGER :: Grades(300)
  INTEGER :: ClassSize, i
  INTEGER :: Bucket (5)

  READ(*,*)  ClassSize, (Grades(i), i = 1, ClassSize)

  DO i = 1, 5
    Bucket(i) = 0
  END DO
```

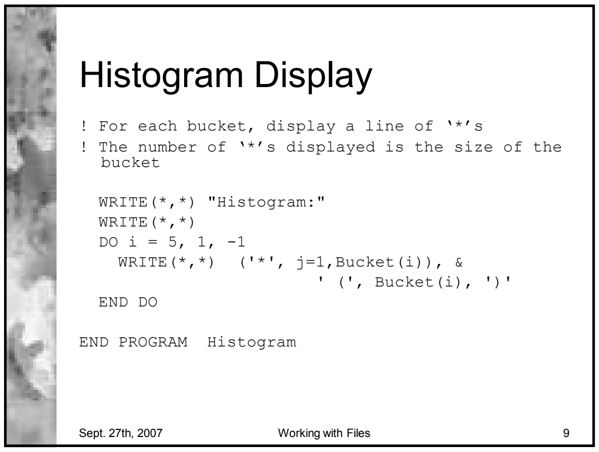
Sept. 27th, 2007 Working with Files 7



Histogram Computation

```
  ! Distribute each grade into the appropriate bucket
  DO i = 1, ClassSize
    IF (Grades(i) < 0 .OR. Grades(i) > 100) THEN
      WRITE(*,*) "Invalid grade for student", i
    ELSE IF (Grades(i) <= 49) THEN
      Bucket(1) = Bucket(1) + 1
    ELSE IF (Grades(i) <= 54) THEN
      Bucket(2) = Bucket(2) + 1
    ELSE IF (Grades(i) <= 64) THEN
      Bucket(3) = Bucket(3) + 1
    ELSE IF (Grades(i) <= 79) THEN
      Bucket(4) = Bucket(4) + 1
    ELSE
      Bucket(5) = Bucket(5) + 1
    END IF
  END DO
```

Sept. 27th, 2007 Working with Files 8



Histogram Display

```
! For each bucket, display a line of ‘*’s
! The number of ‘*’s displayed is the size of the
  bucket

  WRITE(*,*) "Histogram:"
  WRITE(*,*)
  DO i = 5, 1, -1
    WRITE(*,*)  ('*', j=1,Bucket(i)), &
                     ' (', Bucket(i), ')'
  END DO

END PROGRAM  Histogram
```

Sept. 27th, 2007 Working with Files 9

Using Files

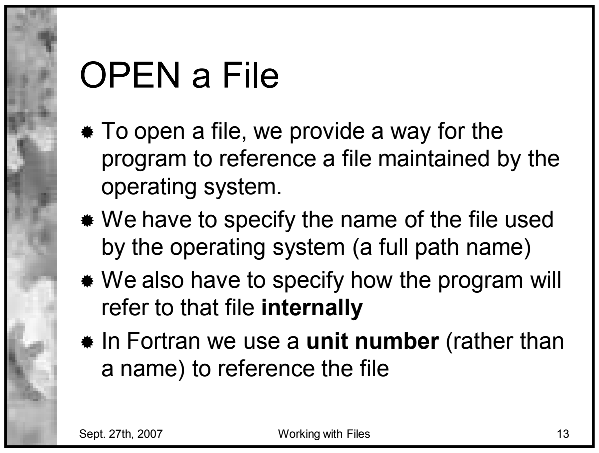
- It's a lot of work to enter the grades for a large class
- It's also very prone to errors
- These values are often generated by other programs such as spreadsheets or by word processors and stored in files
- We would like to read the values directly from these files and be able to write them to other files

File Input and Output

- `READ(*,*) and WRITE(*,*)` read from and write to the standard input (keyboard) and output (screen) devices.
- To read from a file, we have to specify the name of the file and give the program some way of identifying it
- We use this identification to refer to the file in the program

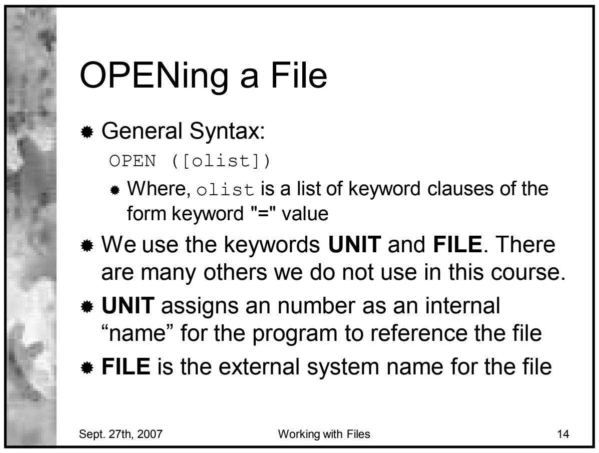
File input/output

- Three steps are required in using a file
  1. Open the file
  2. Input/output using READ and WRITE
     - READ: read data from the opened file
     - WRITE: write data to the opened file
  3. Close the file
     (A file that has not been closed can usually not be accessed afterwards.)



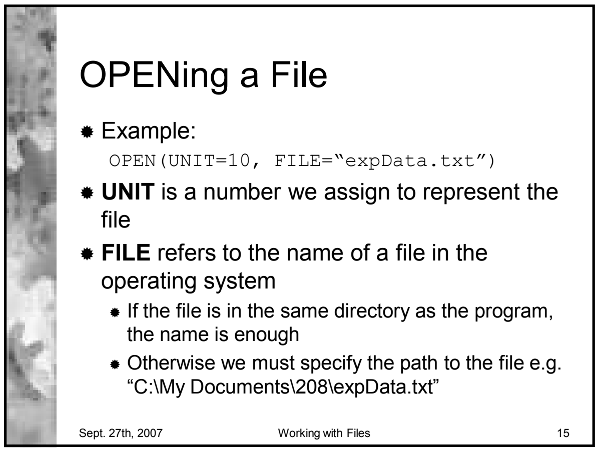
OPEN a File

- To open a file, we provide a way for the program to reference a file maintained by the operating system.
- We have to specify the name of the file used by the operating system (a full path name)
- We also have to specify how the program will refer to that file **internally**
- In Fortran we use a **unit number** (rather than a name) to reference the file



OPENing a File

- General Syntax:
  `OPEN ([olist])`
  - Where, `olist` is a list of keyword clauses of the form keyword "=" value
- We use the keywords **UNIT** and **FILE**. There are many others we do not use in this course.
- **UNIT** assigns an number as an internal “name” for the program to reference the file
- **FILE** is the external system name for the file



OPENing a File

- Example:
  ```
  OPEN(UNIT=10, FILE=“expData.txt”)
  ```
- **UNIT** is a number we assign to represent the file
- **FILE** refers to the name of a file in the operating system
  - If the file is in the same directory as the program, the name is enough
  - Otherwise we must specify the path to the file e.g. “C:\My Documents\208\expData.txt”

FILE input/output

- Once we have opened a file we can read the data that was stored there or we can output data to the file.
- We use the internal unit number to reference the file

```
READ(unit, *) …
WRITE(unit, *) …
```

Histogram File Input

```
OPEN(unit=13,file="histogramdata.txt")
READ(13,*)  ClassSize, &
           (Grades(i), i = 1, ClassSize)
CLOSE(unit=13)
```

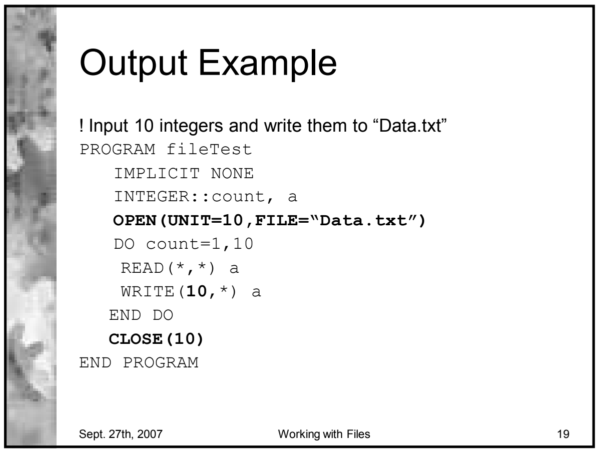
CLOSE

- When we finish using the file we must close it. A file that has not been closed can usually not be read again.
- Syntax:

```
CLOSE ([UNIT=]u)
```

- For example:

```
CLOSE (10)
CLOSE (UNIT=10)
```

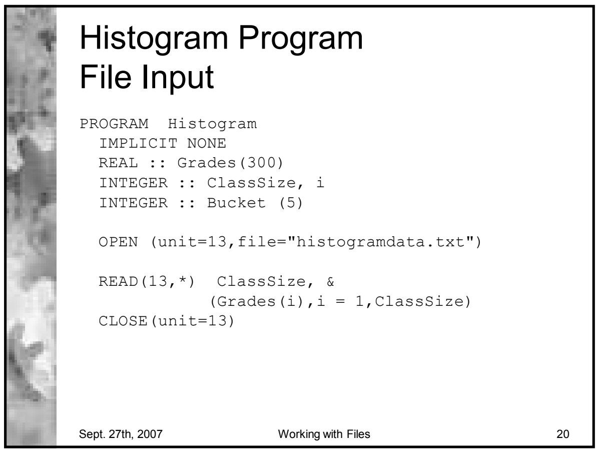


Output Example

! Input 10 integers and write them to “Data.txt”

```
PROGRAM fileTest
    IMPLICIT NONE
    INTEGER::count, a
    OPEN(UNIT=10,FILE=“Data.txt”)
    DO count=1,10
     READ(*,*) a
     WRITE(10,*) a
   END DO
   CLOSE(10)
END PROGRAM
```

Sept. 27th, 2007 Working with Files 19



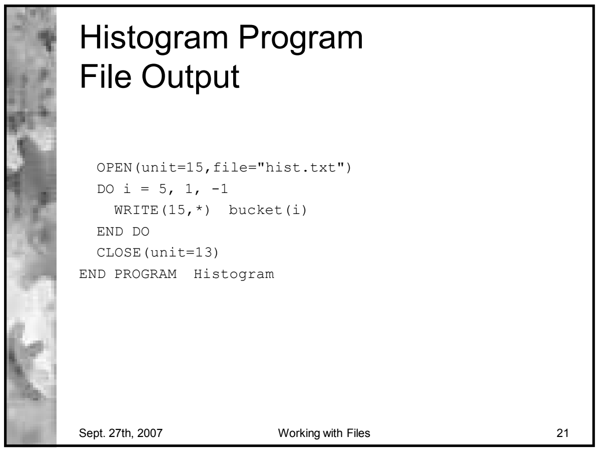
Histogram Program File Input

```
PROGRAM  Histogram
  IMPLICIT NONE
  REAL :: Grades(300)
  INTEGER :: ClassSize, i
  INTEGER :: Bucket (5)

  OPEN (unit=13,file="histogramdata.txt")

  READ(13,*)  ClassSize, &
             (Grades(i),i = 1,ClassSize)
  CLOSE(unit=13)
```

Sept. 27th, 2007 Working with Files 20

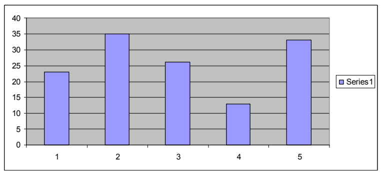


Histogram Program File Output

```
  OPEN(unit=15,file="hist.txt")
  DO i = 5, 1, -1
    WRITE(15,*)  bucket(i)
  END DO
  CLOSE(unit=13)
END PROGRAM  Histogram
```

Sept. 27th, 2007 Working with Files 21

Histogram Using Excel


Series1