CSV FILES — Exam Cheat Sheet

Class XII Computer Science (083) | csv module, reader, writer, delimiters and every standard function

1. What and why

- CSV means Comma Separated Values. It is a plain text file in which one line is one record and the fields of that record are separated by a delimiter, normally a comma.
- It can be opened in Notepad and in Excel, so it is the usual way of moving data between programs.
- Python handles it with the csv module, so every program starts with: import csv

```
E_code,E_name,Scale,Salary      <-- header row (optional)
A01,Bijesh Mehra,S4,65400       <-- one record per line
B02,Vikram Goel,S3,60000
```

2. Opening a CSV file

```
f = open("record.csv", "r")                    # reading
f = open("record.csv", "w", newline="")        # creating (erases file)
f = open("record.csv", "a", newline="")        # appending
```

newline="" is compulsory while writing. Without it Windows inserts a blank line after every record.

Mode	Use it when	Effect on an existing file
"r"	reading, searching, counting	unchanged
"w"	creating the file for the first time	ERASES everything
"a"	adding one more record	keeps old records, adds at the end

3. Writing — csv.writer

Statement	What it does
W = csv.writer(f)	creates a writer object attached to file f
W.writerow(list)	writes ONE record (one list) as one line
W.writerows(listoflists)	writes MANY records in one statement
csv.writer(f, delimiter="*")	uses * instead of a comma as separator

```
import csv

def createFile():
    f = open("StudentActivity.csv", "w", newline="")
    W = csv.writer(f)
    W.writerow(["Student_ID", "Student_Name", "Activity", "Points"])
    W.writerow([101, "Aarav", "Football", 85])
    W.writerow([102, "Ananya", "Basketball", 92])
    f.close()

createFile()
```

4. Reading — csv.reader

Statement	What it does
R = csv.reader(f)	creates a reader object; looping over it gives one record at a time
next(R)	skips the header row (call it once, before the loop)
for row in R:	row is a LIST of the fields of one record
row[0], row[1], ...	the individual fields, all of them strings

Every value read from a CSV file is a string. Before comparing or adding numbers you must convert: int(row[3]) or float(row[3]).

```
import csv

def display():
    f = open("record.csv", "r")
    R = csv.reader(f)
```

```
    next(R)                         # skip the header
    for row in R:
        print(row)                  # row is a list, e.g. ["A01","Bijesh","S4","65400"]
    f.close()

display()
```

5. The five functions asked in exams

(a) Accept / ADD — append one record entered by the user

```
import csv

def Accept():
    f = open("Result.csv", "a", newline="")
    W = csv.writer(f)
    st_id     = int(input("Enter Student ID: "))
    st_name   = input("Enter Student Name: ")
    game_name = input("Enter Game Name: ")
    result    = input("Enter Result: ")
    W.writerow([st_id, st_name, game_name, result])
    f.close()

Accept()
```

(b) Count records that satisfy a condition

```
def wonCount():
    f = open("Result.csv", "r")
    R = csv.reader(f)
    next(R)
    count = 0
    for row in R:
        if row[3] == "Won":
            count += 1
    print(count)
    f.close()
```

(c) Count the total number of records

```
def COUNTR():
    f = open("record.csv", "r")
    R = csv.reader(f)
    next(R)                         # leave this out if there is no header
    count = 0
    for row in R:
        count += 1
    print("Total records:", count)
    f.close()
```

(d) Search — display records above a value

```
def search():
    f = open("furdata.csv", "r")
    R = csv.reader(f)
    next(R)
    for row in R:
        if float(row[2]) > 10000:     # convert before comparing
            print(row)
    f.close()
```

(e) Maximum / minimum, and totals

```
def maxsalary():
    f = open("record.csv", "r")
    R = csv.reader(f)
    next(R)
    max = 0                           # start from 0, not from nothing
    rec = []
    for row in R:
        if int(row[3]) > max:
            max = int(row[3])
            rec = row
    print("Row with the highest salary :", rec)
    f.close()

def CalculateTotalSales():
    total = 0.0
    with open("Sales.csv", "r") as f:
        R = csv.reader(f)
        next(R)
        for row in R:
```

```
            total += int(row[2]) * float(row[3])
    print("Total Sales is:", total)
```

6. A different delimiter (* or ; or tab)

The delimiter must be given BOTH while writing and while reading, otherwise the whole line comes back as one field.

```
Student_ID*Student_Name*Activity*Points
101*Aarav*Football*85
104*Siya*Football*95
```

```
import csv

def createFile():
    f = open("StudentActivity.csv", "w", newline="")
    W = csv.writer(f, delimiter="*")          # * while writing
    W.writerow(["Student_ID", "Student_Name", "Activity", "Points"])
    W.writerow([101, "Aarav", "Football", 85])
    W.writerow([104, "Siya", "Football", 95])
    f.close()

def highScorers():
    f = open("StudentActivity.csv", "r")
    R = csv.reader(f, delimiter="*")          # * while reading too
    next(R)
    for row in R:
        if int(row[3]) >= 90:
            print(row[1], row[2], row[3])
    f.close()

createFile()
highScorers()
```

7. DictReader and DictWriter (fields by name)

```
import csv

# writing
fields = ["ID", "Name", "Marks"]
f = open("data.csv", "w", newline="")
W = csv.DictWriter(f, fieldnames=fields)
W.writeheader()
W.writerow({"ID": 1, "Name": "Aarav", "Marks": 85})
f.close()

# reading — no next() needed, the header becomes the keys
f = open("data.csv", "r")
R = csv.DictReader(f)
for row in R:
    print(row["Name"], row["Marks"])
f.close()
```

8. Text file vs CSV file vs binary file

Point	Text file	CSV file	Binary file
Module needed	none	csv	pickle
Readable in Notepad	yes	yes	no
Extension	.txt	.csv	.dat / .bin
Data read back as	string	list of strings	the original object
Type conversion needed	yes	yes	no
newline="" while writing	no	yes	no

9. Marks are lost here — check every time

- Forgetting import csv.
- Forgetting newline="" while opening for writing — a blank line appears between records.
- Using "w" when the question says add a record — it deletes the whole file. Use "a".
- Forgetting next(R) and then trying int() on the header row — it raises a ValueError.
- Comparing without converting: row[3] > 1000 is wrong because row[3] is a string. Use int(row[3]) > 1000.
- Writing Next(R) with a capital N. The function is next(R), all lowercase.
- Giving the delimiter while writing but not while reading, or the other way round.
- writerow() needs a list. Passing four separate values instead of one list is an error.

- Not closing the file, or not calling the function at the end of the program.