HP-12C Quick Reference

Memory & Display

Memory	Stack, Last-X, 5 financial registers, 20 storage registers, initially 1 program register worth about 8 program instructions. Program memory shared with storage registers
MEM	Displays memory usage in the form "`P-xx r-nn`" where: xx: The number of program lines (up to 99) nn: The number of available storage registers
STO 0-9, .0-.9	Store X in specified storage register
STO +−x÷ 0-4	Register store arithmetic: Register OP X → Register
RCL 0-9, .0-.9	Recall specified storage register into X. Register recall arithmetic is not supported
f n	Choose fix point format with n digits after the decimal point (same as FIX n on other calculators)
f .	Choose exponential format with a maximum of digits (same as SCI 9 on other calculators)
Decimal separator	Turn off, press & hold ON, press and hold ".", release ON release "." to toggle between a dot and a comma for the decimal separator

Clearing Data

CLx	Clear X register. It is not possible to clear individual digits of a number that is currently being entered
CLEAR Σ	Clear statistics registers R1-R6 and the stack
CLEAR PRGM	PRGM mode: Clear all program memory RUN mode: Set program counter to 00
CLEAR FIN	Clear financial registers
CLEAR REG	Clear storage registers including statistics registers, financial registers, stack and LastX
CLEAR PREFIX	Clear prefix key and briefly display all 10 digits of X

Financial Functions

%	Percent. Note that the stack doesn't drop: base number Y is preserved
Δ%	Percentual difference from Y to X. Base number Y is preserved
%T	Calculates what percentage of Y is X. Base number Y is preserved

Financial registers

n: Number of compounding periods. For monthly payments this is the number of years multiplied by 12.
When specified this is usually an integer value. If it is fractional then an "odd first period" is assumed where the interest begins to accrue *before* the first regular period. Ie. if n=10.5 then interest starts to accrue half a period before the first regular period.

	When calculated the result is usually not an integer number of periods. However, the result will always rounded up to the next integer value if the fractional part is >0.005
i:	Interest rate per compounding period in percent. For monthly compounding this is the annual interest rate (or annual percentage rate, APR) divided by 12
PV:	Present value of asset or loan
PMT:	Amount payed or received per compounding period
FV:	Future value of asset or loan at the end of n compounding periods

Compound Interest Calculations	
Enter values	Simply press the financial register key. This will store X in the associated register – as long as *no other* financial register key (including 12x & 12÷) has been pressed just before this one
Calculate unknown value	If after a financial register key has been pressed to enter some data (including 12x & 12÷) a 2nd financial register key is pressed (the same or a different one) the corresponding value will be calculated depending on the values of the other registers using *compounding* interest calculations.
	Note that *any* of the values n, i, PV, PMT and FV can be calculated as long as the remaining four other values have been specified!
	When a value has been calculated, it is displayed in the X-register and also stored in the corresponding financial register.
	Cash flow sign convention: • Amounts received are positive • Amounts payed out are negative
	Example 1: Savings plan
	PV: Current savings of –1000 (negative because this amount has initially been payed out)
	i: Interest rate: 2.5% annually, quarterly compounded → 0.625% per compounding period
	PMT: Monthly payment –100 (negative because payed out) → –400 per compounding period
	n: Number of compounding periods: 10 years → 40 periods
	FV: Result: After 10 years the savings added up to 19396.74 (positive because this amount is received)
	Example 2: Loan
	PV: Amount lent 9000 (positive because this amount has been received)
	i: Interest rate: 7% annually, monthly compounded → 0.58% per period
	PMT: Monthly payment –250 (negative because payed out)
	FV: 0 because we want to pay beck the entire loan
	n: Result: After 41 periods or 3.42 years the loan has been payed back

Recall values	Press RCL <financial register>. Note that after this the next financial register key press will store the X register contents in the financial register rather than calculating an unknown value
12x	Multiply the value in X by 12 and store in financial register n
12÷	Divide the value in X by 12 and store in financial register i
END	Specify that payments are made at the beginning of the compounding period. Default value
BEG	Specify that payments are made at the beginning of the compounding period. Indicated by "BEGIN" in the display
STO EEX	Toggles the "C" indicator in the display. This command is not programmable. On: Compound interest accrues during odd period Off: Simple interest accrues during odd period
INT	Simple interest on a 365 and 360 day basis Input: n: Number of days i: Annual interest rate in percent PV: Principal amount Output after INT: X: Accrued simple interest on a 360-day basis. Note that according to the cash flow sign convention the result will have the opposite sign of PV Y: –PV, so the total sum (principal + interest) can be calculated by simply pressing "+" Z: Accrued simple interest on a 365-day basis. To add principal and 365-day interest press R↓, then optionally x↔y to view the accrued interest and then +

Discounted Cash Flow Analysis

Supported methods are *Net Present Value (NPV)* and *Internal Rate Of Return (IRR)*.
This is similar to the compound interest calculations above with the exception that the recurring payments PMT (or cash flows) need not be equal. However, the interest rate and the time interval per compounding period must still be constant.

Input:

- i: For NPV only: Interest rate or *(required/minimally acceptable) rate of return* or *cost of capital*.
- CFo: Initial cash flow or payment or present value. This also sets n=1
- CFj: Enters the subsequent cash flows one after the other. The values will be stored in storage registers starting with R1. Each entry increases n by 1
- Nj: Optional. *After* each cash flow entry (thru CFj) enter how often the cash flow recurs. This saves storage registers but can only be used for *consecutive* cash flows of the same amount

Verification:

1. Optionally, enter a specific value for n. By default n is the number of different cash flows that have been entered using CFj/Nj
2. Press RCL Nj (optional) and RCL CFj to review the number of times the cash flow recurs and the cash flow amount. RCL CFj will also decrease n
3. Repeat step 2: Since each RCL CFj decreases n this will display the number of recurrences and the cash flow amounts in reverse order

4. Make sure to set n to the correct value after the verification!

Output:

NPV: This calculates the *net present value* of the future cash flows: So basically, this performs a regular compound calculation with unequal cash flows, then projects the result to the present time by dividing thru $(1+i)^n$ which is essentially a "reverse compound calculation" and finally sustracts the initial payment CFo.

Or in other words: The investment performs as good as if an amount of NPV was compounded over n periods at an interest rate of i with no periodic cash flows.

Conceptually, the calculation of NPR assumes that CFo is a loan that comes at an interest rate of i. Subsequent cash flows revenues are applied towards the loan; subsequent cash flow investments are financed at the same interest rate than the initial loan.

See Owner's Handbook pg.167 for the calculation of the *Modified internal rate of return (MIRR)* which allows for different interest rates for investments and payouts.

NPV>0: The investors financial assets are increased and the investment is attractive.

NPV<0: The investors financial assets are decreased and the investment is not attractive.

The result is stored in PV

IRR: Calculates the *internal rate of return*: This is the rate of return (percentage) at which NPR yields a value of 0.

If the cash flows were all identical the resulting interest rate would be *exactly* the same as for a regular compounding calculation with FV=0 that was solved for the interest rate i.

NOTE: The calculation of IRR can have as many valid solutions as the sign of the cash flows changes over time!

The result is stored in i

Amortization (AMORT)

Calculates the amounts applied toward principal and interest from a single loan payment or multiple payments.

All calculated values are rounded to the display's number of fractional digits.

Input:

i: Interest rate per compounding period

PV: Amount of the loan (=principal)

PMT: Periodic payment (must be negative)

X: Number of payments

Output after AMORT:

X: Amount from those payments applied towards interest

Y: Amount from those payments applied towards the principal

PV: Remaining balance after the given number of payments. This number decreases with subsequent AMORT calculations

n: Total number of payments amortized. This number increases with subsequent AMORT calculations

Depriciation Calculations

Input:

- PV: Original cost of asset
- FV: Salvage value of asset, ie. 0
- n: Expected useful life of asset in years
- i: If "declining-balance" (DB) method is used, enter declining balance factor as a percentage. Ie. 1¼ times straight-line rate is entered as 125
- X: Number of the year for which depriciation is to be calculated, starting with 1 for the first year

Output:

- SL: Return depriciation for the specified year using the *straight line* method.
  The depriciation is constant: D=(PV-FV)/n
- SOYD: Calculate depriciation using the *sum-of-the-years-digits* method
- DB: Calculate depriciation using the *declining-balance* method. The depriciation is a percentage of the current book value BV: D = BV * i / (100*n)
  Note that after n years of depriciation the remaining value RV is not the salvage value but: RV = PV * (1 – i/(100*n)^{n}
  For this reason the depreciation method is often changed from DB to SL at a certain point in time
- Y: Remaining value of the asset after depreciation. PV is not affected!

See Owner's Handbook section 13 (pg.153) when the acquisition date does not coincide with the fiscal year or when the deprociation method is changed from DB to SL

PRICE, BOND	Bond calculations, see pg.76ff of the Owner's Handbook

Calendar Functions

General	Calendar functions can handle dates from 15.10.1582 to 25.11.4046
M.DY	Choose display format `mm.ddyyyy` (US format)
D.MY	Choose display format `dd.mmyyyy` (European format, indicated by D.MY in the display)
DATE	Adds a number of days in X to a date in Y. The number of days is substracted if X<0. The result is displayed in the following form: `dd.mm.yyyy w` Where w is the weekday of the date 1…7 with 1=Monday
∆DYS	Calculates the number of days between two dates in X and Y. X: Number of days between X & Y Y: Number of days between dates X & Y on the basis of a 30-day month

Statistics Functions

Memory	Statistics registers: R1=n R2=Σx R3=Σx² R4=Σy R5=Σy² R6=Σxy
Σ+	Add values to the above registers and increment n
Σ-	Substract values from above registers and decrement n
$\bar{x}$	Calculate ∑x & ∑y mean value and place result in X & Y. Requires n>0
s	Calculate ∑x & ∑y standard deviation and place result in X & Y. sx=SQRT[{n∑x^2 – (∑x)^2} / {n(n-1)}] Requires n>1
$\bar{y}$,r	This function assumes a straight line thru the (X,Y) data points and calculates for a given X the approximated $\bar{y}$ value which is returned in X. In Y this function returns an estimate how close the data points come to a straight line. +1 indicates that all points lie on a line with positive slope, -1 indicates that all points lie on a line with negative slope, 0 indicates that an approximation by a straight line isn't possible. Requires n>1
$\bar{x}$,r	Same as above but for a given y an estimated x is calculated
$\bar{x}$,w	Use ∑+ with the value in Y and its weight in X. Calculates weighted mean w= ∑xy / ∑x
n!	Faculty of X. Only integer X are allowed

Miscellaneous Functions

RND	Round X to the current number of displayed fractional digits
FRAC	Return fractional value of X
INTG	Return integer value of X
y^x	Y to the power of X. Y must be positive

Programming

Memory	Initially, there are only 8 progam lines available. As more are needed storage registers are converted to program space, each registers allowing for 8 more program lines. Registers 0-6 will never be converted to program space. A maximum of 99 program lines can be entered
P/R	Switch between PRGM and RUN mode
Instruction codes	Instructions will be inserted *after* the currently displayed line. Instructions are displayed by their row/column code with numbers having their special codes 00 to 09 and the 10th column having code 0 (ie. code 40 corresponds to "+")
Inserting and deleting instructions	*Not possible!* It is only possible to overwrite instructions. A newly entered instruction will overwrite the one after the currently displayed instruction. To insert instructions in a lengthy program enter a GTO to the end of the porgram space, add instructions there (including the overwritten one) and then jump back to the original location

GTO nn	PRGM mode: Insert jump to specified program line. Note that GTO 00 will always stop the program RUN mode: Set program counter to specified line *There are neither symbolic labels nor subroutine calls available!!*
GTO . nn	PRGM & RUN mode: Set program counter to specified line
R/S	Run or stop program
SST	PRGM mode: Step forward thru program lines RUN mode: Display next program line while held down then execute code
BST	PRGM mode: Step backward thru program lines RUN mode: Display previous program line while held down then set program counter back to previous line but do not execute code
PSE	Halt program execution for about 1 second and display X register
X≤Y, X=0	Relational operations: If relation is true next program line is exectued If relation is false next program line is skipped