Printing Financial Calculator Sets New Standards for Accuracy and Capability

This briefcase-portable calculator has several new functions and is exceptionally easy to use. Most important, the user need not be concerned about questions of accuracy or operating limits.

by Roy E. Martin

HEWLETT-PACKARD INTRODUCED its first financial calculator, the HP-80, in 1973.[1] The HP-80 was followed, although never replaced, by the HP-81, the HP-70, the HP-22,[2] and the HP-27.

The new HP-92 Financial Calculator, Fig. 1, while superficially similar in many respects to these units, vastly exceeds all of them in functional capability and accuracy. Originally conceived as a briefcase-portable printing calculator packaged like the HP-91[3] and the HP-97[4] and having the financial capabilities of the HP-22, the HP-92 in reality goes far beyond this modest goal. Among its features are:

- Compound interest keys redefined to enhance capability and ease of use
- A printed amortization schedule, correctly rounded and clearly labeled
- Internal rate of return (IRR) that allows the user to enter up to 31 cash flows with arbitrary positive and negative values
- The greatest accuracy ever achieved in any HP financial calculator
- Calendar functions with a range of 900,000 days (approximately 2464 years)
- Bond and note functions that conform to Securities Industry Association equations[5]
- Three types of depreciation that can be done after entering data only once
- Means, standard deviations, and linear regression for two variables.

New Compound Interest Keys

The cornerstone of the HP-80 and all subsequent HP financial calculators is the row of compound interest keys: **n** **i** **PV** **PMT** **FV**

n = number of compounding periods

i = percent interest per period

PV, **PMT**, **FV** specify the cash values in various problems (**PV** = present value; **PMT** = payment; **FV** = future or final value).

These keys allow the user to solve for an unknown value by first placing known values in the calculator and then pressing the key corresponding to the unknown.

Example: Find the monthly payment due on a 36-month, 12%, $3000 loan.

	Keystrokes		
These keystrokes	**36**	**n**	
place the known	**1**	**i**	(12% annual is 1% per month)
values into the	**3000**	**PV**	
calculator			
Then press:		**PMT**	
Answer displayed:		**99.64**	Monthly Payment

This sequence of keystrokes will solve this problem on all previous HP financial calculators.*

The compound interest keys solve three types of problems, based on the following three equations. (In these and subsequent equations, i is a decimal fraction, e.g., 0.05 for five percent.)

$FV = PV(1+i)^n$	Compound Amount
$PV = PMT[1-(1+i)^{-n}]/i$	Loan
$FV = PMT[(1+i)^n-1]/i$	Sinking Fund

Each of these equations has four variables. As long as three of the four variables are known (n or i must be one of the three knowns) a user can solve for an unknown.

Because there are three distinct equations and only one set of keys, it is necessary to specify which equation is involved. This is done automatically through the use of status bits (flags). Internally, status bits are set when values associated with n, i, PV, PMT, FV are keyed into the calculator. As soon as three status bits are set, the equation is specified and a value can be computed.

On the HP-80, known values are pushed onto the stack and then lost when a value is computed, requiring the reentry of data on every new computation. The HP-70, HP-22, and HP-27 have separate registers to hold the financial values but require special functions to clear the status bits.

*The HP-27 requires the use of a shift key but is fundamentally the same.

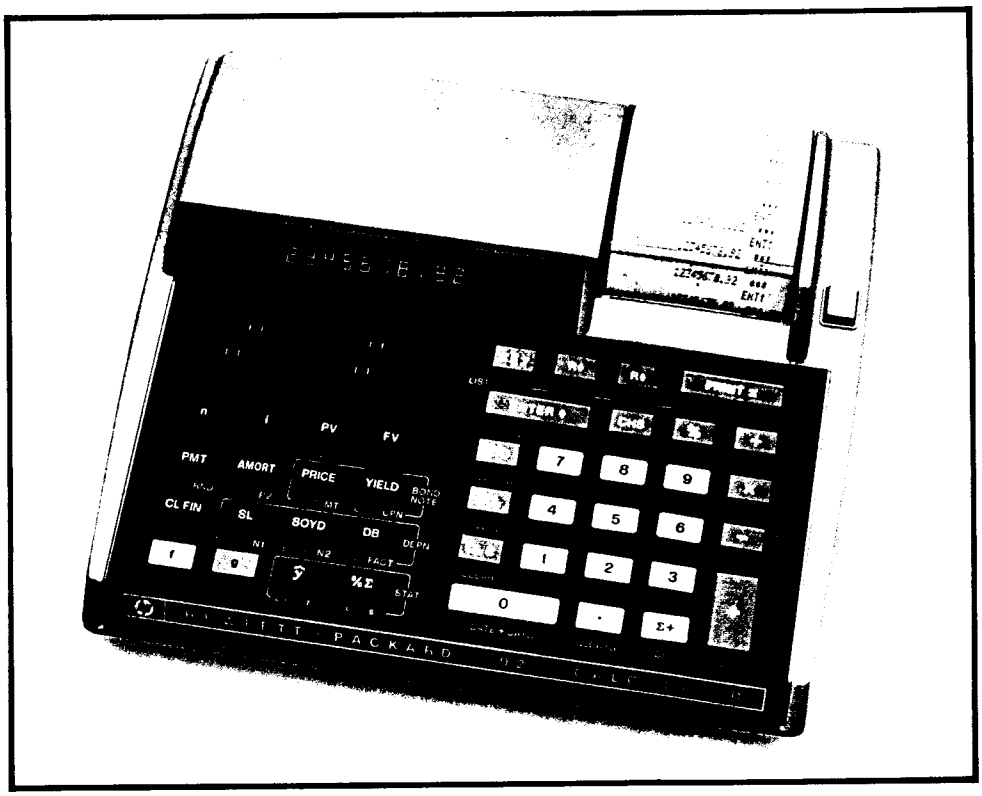

Fig. 1. *HP-92 Investor is a financial printing calculator with superior accuracy and capability. Keyboard is designed to prompt the user, making many problem solutions obvious even without a manual.*

This design, although creatively conceived and cleanly implemented, is inconvenient for chained calculations. Also, an important class of problems, loans with a balance, cannot be solved without tedious iteration by the user.

The same keys, n, i, **PV**, **PMT**, **FV**, were to be on the HP-92. However, we wanted to improve and simplify their use. The most attractive alternative came in the form of a more general equation:

$$PV\,(1+i)^n + PMT\,[(1+i)^n-1]/i + FV = 0.$$

The three equations in previous calculators are all special cases of this one, up to a sign change. The basic premise in this equation and a major difference between the HP-92 and other financial calculators is that money paid out is considered negative and money received is considered positive.

Implemented in the HP-92, this equation allows free-format problem solving, letting the user change any variable at any time or solve for any value at any time. It also increases the functional capability of the calculator to include loans with a balance, fixes the roles of PV, PMT, and FV, making them easier to explain, reduces the number of equations from three to one, and eliminates the need for status bits—the data in the calculator determines the problem to be solved.

In the early stages of the project, the new compound interest equation was simulated. The increase in capability and simplicity was substantial. Within minutes, inexperienced people could understand the concept and apply the keys to problems formerly considered too complicated to solve. Naturally, we were pleased. The new calculator would be more capable than earlier designs and easier to use as well. But our satisfaction was short-lived, for it turned out that here,

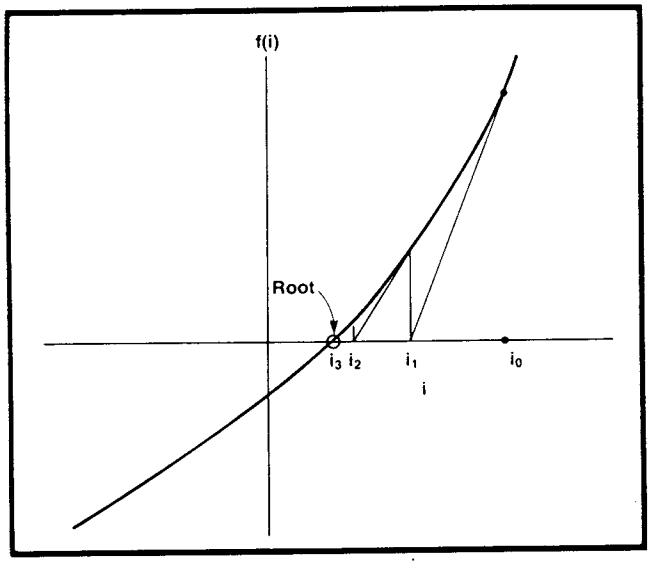

Fig. 2. *Newton's method is used by the HP-92 to solve compound interest problems. Starting from some point i_0 on the graph of an equation, the goal is to find the root of the equation, or the point where the graph crosses the axis. Drawing a tangent line to the graph at i_0 and finding where this line crosses the axis gives a second point i_1. This process is repeated to find i_2, i_3, and so on, until a point is reached that is close enough to where $f=0$. i_0 is called the initial guess.*

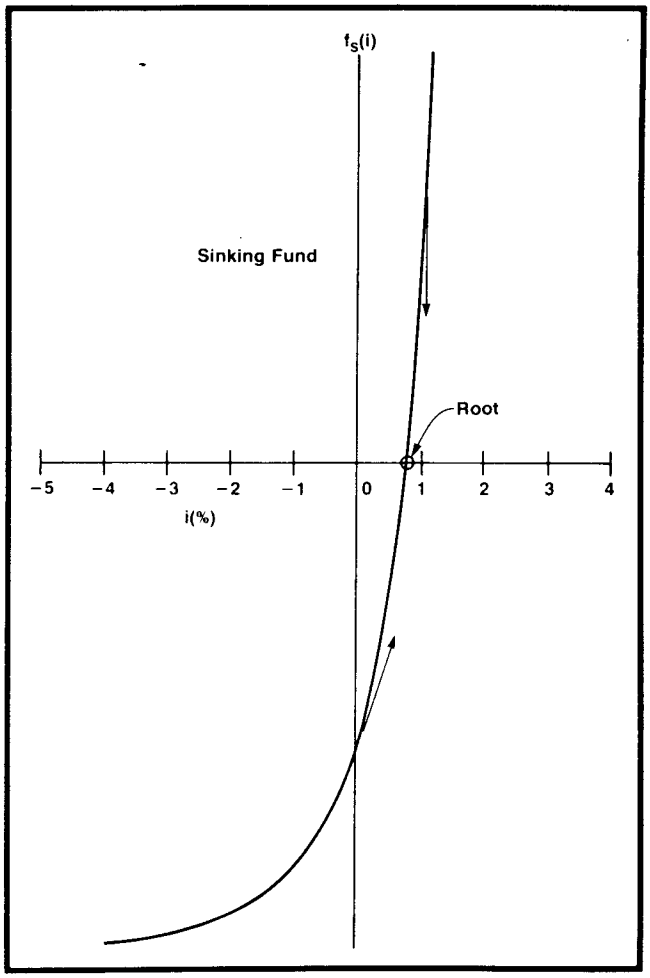

Fig. 3. *Equations used in previous HP financial calculators have favorable graph shapes (the one shown is typical), so that starting from any initial guess i_0 the steps taken by Newton's method are always toward the root.*

as usual, nothing is free.

The numerical analysis used in solving the three equations in the HP-80 had been formidable. Yet the accuracy and reliability of the algorithms was borderline and their performance deteriorated unacceptably when they were applied to the new more general equation. The most difficult problem was solving for i in the compound interest problems. Internally, this involves the microprogrammed application of Newton's method in the solution of polynomial equations (see Fig. 2).

Newton's method requires an initial guess, i_0, at the root of f(i)=0. Subsequent values are produced using

$$i_{k+1} = i_k + \frac{f(i_k)}{f'(i_k)}$$

until $|i_k - i_{k+1}| <$ required error limit. Basically, we slide down the graph of f(i) sawtoothing into the solution.

Three factors that affect the use of Newton's method are the shape of the graph, the accuracy of evaluation of the function f(i) and its derivative, and the quality of the initial guess. For certain graphs any reasonable initial guess will produce convergence to the correct answer. This was the case with the equations solved by previous HP financial calculators (see Fig. 3).

Inaccuracy in evaluation of the function and its derivative can cause various problems. For example, a small error can cause the iteration to step in the wrong direction, say to the previous point, resulting in an infinite loop. Worse yet, it can produce a wrong answer. The new more general equation was more sensitive than the old to round-off errors, and introduced another difficulty not encountered before.

The quality of the initial guess became a critical issue. Unless the initial guess was good enough, Newton's method would fail (see Fig. 4). With this in mind, we implemented several transformations to change the shapes of the graphs in an attempt to make Newton's iteration less sensitive to poor first guesses. We also carried extra digits and programmed numerically stable formulas to diminish the impact of rounding errors on the accuracy of intermediate calculations.

But our work was far from done. Even with the transformations and increased accuracy, initial guesses in error by less than 1% proved inadequate, because convergence was too slow when n was large.

After four months of careful examination and simu-

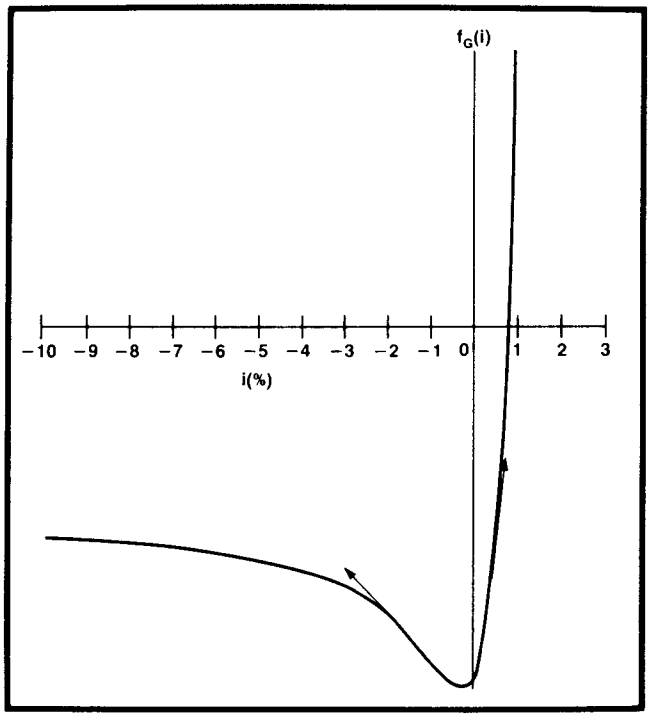

Fig. 4. *Modified equation used in HP-92 enhances ease of use, but is more difficult to solve. Shape of graph is such that some initial guesses will cause Newton's method to step away from the root. To prevent this a strategy was developed that produces initial guesses accurate to five decimal places.*

Using the n, i, PV, PMT, FV Keys

Corresponding to each of these keys is a storage register. To put a value in the storage register, just key in the value and then press the appropriate key. Money paid out is represented as negative and money received is represented as positive.

Problem:

1. If you deposit $10,000 in a fund that pays 7.75% annual rate, how much could you withdraw 12 years later?
2. If, in addition, you deposit $1000 each year thereafter, how much would you be able to withdraw after 12 years?
3. If you wanted to withdraw $45,000 at the end of the 12-year period, how much would you have to deposit each year?.
4. If you could deposit $18,500 initially, how much would you have to deposit each year to be able to withdraw $45,000 at the end of the 12 years?

Solution:

Press CL FIN . This clears the registers.

1. Key In / Then Press / Comment

Key In	Then Press	Comment
12	n	This is the number of years.
7.75	i	This is the periodic interest rate.
10,000 CHS	PV	You are putting the money into the bank so you key it in as negative.
	FV	This tells the calculator that you wish to solve for the cash flow at the end of the time period.

See displayed: 24,491.05, the amount you could withdraw in 12 years.

2. After values are keyed in (or calculated), they remain in the registers. To do the second part of the problem, all we have to do is key −1000 into PMT (12 remains in n, 7.75 in i and −10,000 in PV) and then press FV

Key In	Then Press	Comment
1000 CHS	PMT	Again payment is negative because you are giving money to the bank.
	FV	This tells the calculator to find the cash flow at the end of the 12 years.

See displayed 43,189.17. The amount you could withdraw after 12 years.

3. If you needed to withdraw $45,000 and wanted to find out what your yearly deposit would be, put 45,000 into FV and then tell the calculator to solve for PMT

Key In	Then Press	Comment
45,000	FV	At the end of the 12 years you will receive $45,000.
	PMT	This tells the calculator to find the annual deposit you must make.

See displayed −1096.85. The amount you must deposit annually.

4. Now put −18,500 into PV, then press PMT

Key In	Then Press	Comment
18,500 CHS	PV	You plan to deposit $18,500 at the beginning of the 12 years.
	PMT	What will your deposit be so that you can still withdraw $45,000 at the end of 12 years?

See displayed 16.50 This tells you that you could *withdraw* this amount each year and still get $45,000 at the end of 12 years.

Fig. 5. *An example illustrating how natural the HP-92's compound interest keys are to use. An important difference from previous financial calculators is that money paid out is considered negative and money received is considered positive.*

lation we devised an initial guess strategy that produces guesses correct to five places over all ranges of PV, FV, PMT, and i, and with n as large as 10^8. Computation time for i was reduced to about a dozen seconds.

Some of the techniques employed were:

- An initial guess strategy that selects an initial guess by problem classification, the production of as many as three guesses, and the selection of the final initial guess based upon the three guesses
- Enhanced accuracy in $+, -, \times, \div, \ln, e^x$
- Special evaluation of $[(1+i)^n-1]/i$ to avoid damage from cancellation
- Carrying more digits internally than any previous HP financial calculator.

In the final implementation of the **n**, **i**, **PV**, **PMT**, and **FV** keys we were able to achieve reliable functional capability over a wide range of data and problems, a dramatic enhancement in ease of use, and definitive accuracy (see accuracy discussion) exceeding that of any previous HP calculator.

Fig. 5 demonstrates how easy the new compound interest keys are to use.

Internal Rate of Return

Given an initial investment and a series of uneven cash flows CF_0, CF_1,..., CF_n occurring at equally spaced time intervals the IRR (internal rate of return) is the interest rate that satisfies the following equation:

$$CF_0+CF_1(1+i)^{-1}+CF_2(1+i)^{-2}+...+CF_n(1+i)^{-n}=0.$$

The only other HP financial calculators to produce IRR are the HP-27, which allows eleven cash flows, and the HP-81, which allows ten cash flows. The HP-92 allows up to 31 uneven cash flows.

We again applied Newton's method to solve this equation, but in this case the shape of the graph presented a different type of problem. In the compound interest problem there is only one root (the graph crosses the axis only once). In the IRR problem it is possible for the equation to have many roots. Descartes' rule of signs allows polynomial equations with several changes of sign in their coefficients to have several roots. Since the cash flows in the IRR problem represent the coefficients of a polynomial (see equation), cash flows that change direction more than once produce this possibility. However, if there is more than one root, none of the solutions will be financially meaningful. To avoid this complication, the HP-27 will not allow more than one sign change.*

Example: Consider the following two problems. Negative values represent investment and positive values represent income.

	Problem 1	Problem 2
Initial	−$10,000	−$10,000
Year 1	−$ 1,000	$ 2,000
Year 2	$ 2,000	−$ 1,000
Year 3	$13,000	$13,000

The HP-27 produces an answer of 11.83% for Problem 1 but returns **ERROR** for Problem 2. To most users it is not apparent why this happens.

We wanted to remove this kind of limitation. Again

*It should be noted here that the techniques used in the HP-27 were the best available at the time. Many implementations of IRR take no precautions to protect the user from anomalous answers.

after considerable investigation we were able to implement an IRR function with a much broader range. For Problem 2 above the HP-92 produces the correct answer of 12.99%.

The IRR function on the HP-92 will produce the correct answer for any problem with up to 31 cash flows and any number of sign changes, provided that there is at least one sign change and that there is only one *significant* sign change. In general, this means that there is only one real root. Multiple sign changes are allowed provided that all but one of the cash flows changing sign are small in comparison to the other cash flows.

Example:

	Problem 3 Acceptable	Problem 4 Unacceptable
Initial	−$100,000.00	−$100,000.00
Year 1	$500.00	$500,000.00
Year 2	−$200.00	−$200,000.00
Year 3	$100.00	$100,000.00
Year 4	$150,000.00	$150,000.00

For Problem 3 the HP-92 produces the correct answer of 10.77%. For Problem 4 the HP-92 will calculate indefinitely. The mathematically correct but financially meaningless answers to Problem 4 are −147.31% and 362.98%.This does not mean that the problem is financially meaningless, but only that IRR is not the way to attack it. If there is a financially meaningful answer to an IRR problem the HP-92 will find it.

Bonds

The SIA (Securities Industry Association) handbook[5] specifies certain procedures for the calculation of bond values. Most bonds have semiannual coupon periods determined by their maturity dates. For example, if a bond matures on December 15, 1985, then the coupon periods will end on June 15, 1985, December 15, 1984, June 15, 1984, and so on. A bond is not usually purchased on a coupon date (see Fig. 6). This implies that both simple and compound interest must be used during calculations of price and yield. The SIA procedure for the calculation of purchase price involves the exact number of days in the coupon period in which the bond is purchased. The number of days in a coupon period can vary from 180 to 184. Inside the HP-92 the calendar functions determine the exact number of days to the end of the coupon period from the purchase or settlement date, automatically taking leap years into account (Fig. 7). The computations can be based on a 360 or 365-day year.

A Manual on the Keyboard

The HP-92's keyboard is designed to prompt the user and make it obvious how to solve many problems. Keys of the same kind are grouped together. In many problems all required input parameters have individual storage registers. To place a value in one of these registers the user simply keys in the value and then presses the key corresponding to that register.

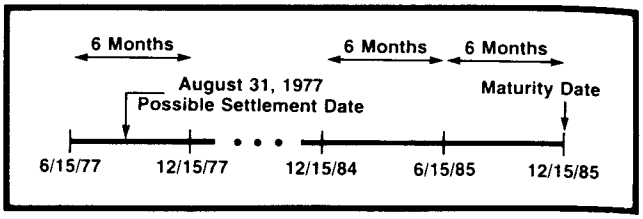

Fig. 6. *In bond calculations, coupon dates are determined by the maturity date and are six months apart. Settlement (purchase) date can be any business day. Built-in HP-92 calendar functions determine the exact number of days between the settlement date and the coupon date.*

Example: There are three types of depreciation: straight line (SL), sum of the years digits (SOYD), and declining balance (DB). The input parameters and the corresponding keys are life (**LIFE**), starting period (**N1**), book value (**BOOK**), ending period (**N2**), salvage value (**SAL**), and declining balance factor (**FACT**). These values are loaded into their registers using the blue and gold shift keys where appropriate. Once this is done, any or all of the three types of depreciation schedules may be calculated by pressing the **SL**, **SOYD**, or **DB** keys.

Accuracy and Operating Limits

Everyone who participated in the HP-92's design wanted to produce a calculator whose reliability, accuracy, and capability would exceed whatever might reasonably be demanded of it. Previous calculators would have to be surpassed, if only because as time passes, users take previous accomplishments for granted and demand more. One target for improvement was accuracy. Consider the following slightly unrealistic problem.

Example: Find the present value and the future value of 63 periodic payments of one million dollars each at the (very tiny but still positive) interest rate i = 0.00000161%.

Problem:
Calculate the price of a corporate bond with a settlement date of August 24, 1977, a maturity date of March 15, 2000, a coupon rate of 8.75% and a yield of 8%. (Calculated on 30-day month, 360 day year.)

Solution:
Enter the settlement date, maturity date, coupon rate, and yield. Press PRICE. The bond's accumulated interest and price are then printed.

8.241977	ST
3.152000	MT
8.750000	CPN
8.000000	YLD
BOND *360	PRC
3.864583	AI
107.768456	***

Fig. 7. *A bond problem and the HP-92 solution. That February has only 28 days is automatically taken into account.*

	HP-80	HP-22,27	HP-92
PV	62,608,695.65	63,000,000.00	62,999,967.54
FV	62,608,695.65	62,981,366.46	63,000,031.44

The HP-92 answers are correct, but more significant, the other answers are clearly wrong: interest is positive but money is lost.

Obvious errors even on such unrealistic problems can undermine user confidence. The only way to prevent apprehension is to preclude all anomalies. For this reason, we set out to produce such robust algorithms that the user need never be concerned with questions of accuracy or operating limits. The extent of our success may be gauged by the reader's readiness to forget the limitations explained below.

Calendar Functions: **IS, ST, MT** Dates of issue, settlement, maturity
Δ DAYS Days between dates
DATE + DAYS
g PRINT x Day of the week.

These functions accept dates from October 15, 1582 to November 25, 4046. The first date marks the inception of the Gregorian calendar, now in use throughout Europe and the Americas, in which leap years are those evenly divisible by 4, but not by 100 unless also by 400. (The year 2000 will be a leap year, but not 1900 nor 2100.) The second date is determined by internal register limitations, not by any special knowledge of the future.

Mathematical Operations: +, −, ×, ÷, 1/x, %, %Σ, Δ%, $\sqrt{x}$, e^x, LN

Error is less than one unit in the last (tenth) significant digit over a range of magnitudes including 10^{-99} and 9.999999999×10^{99}. y^x is also accurate to within one unit in the last significant digit for $10^{-20} \leq y^x \leq 10^{20}$; outside that range the error is less than ten units in the last significant digit.

Statistics: Σ+, Σ−

These keys accumulate various sums using arithmetic to ten significant digits. This determines the range and accuracy achievable by the other statistical keys $\hat{y}$, **LR**, r, $\bar{x}$, and **s**. For x data consisting of four-digit integers, $\bar{x}$ and **s** will be correct to ten significant digits and $\hat{y}$, r, and **LR** will be in error by less than the effect of perturbing each y value by one unit in its tenth significant digit. For x data with more than four digits per point the error can be significant if the data points have redundant leading digits; in this case both time (keystrokes) and accuracy will be conserved if the redundant digits are not entered, following recommendations by D.W. Harms.[6]

Bond Yield and Interest Rates: YIELD, i, IRR.

The error will be smaller than one unit in the last (tenth) significant digit or 0.000000001, provided that the number of periods n does not exceed 1,000,000, and for IRR, provided that the cash flows reverse sign significantly only once as described above. These rates are calculated far more accurately than the Securities Industry Association requires.

Money Values: PRICE, PMT, PV, FV, AMORT, SL, SOYD, DB, n

Errors will be smaller than the effect of changing all input values in their tenth significant digits. Typically, this means that if $(1+i)^n$ does not exceed 1000 then errors will be less than one unit in the last (tenth) digit. This amounts to a fraction of a cent in transactions involving tens of millions of dollars.

Verifying Accuracy

A simple means of verifying the accuracy of a given computation on any calculator is to attempt to recalculate the known quantities using a quantity the calculator has computed based on the knowns.

Example: Key the following values into the HP-92:

FEATURES AND SPECIFICATIONS
HP-92 Investor

MAN ALL NORM Controls printing of keyboard operations.
BEGIN END NOTE BOND Selects payments at beginning or end of period; or selects bond or note calculations.
360 365 Day basis switch for calendar, bond/note, and interest calculations.

COMPOUND INTEREST
n Stores or computes number of periods.
12× Converts number of periods from years to months.
i Stores or computes interest rate per compounding period.
12÷ Converts interest from yearly to monthly rate.
PV Stores or computes present value (initial cash flow at the beginning of a financial problem).
FV Stores or computes future value (final cash flow at the end of a financial problem).
PMT Stores or computes payment amount.

DISCOUNTED CASH FLOW ANALYSIS
NPV Computes net present value of future cash flows.
IRR Computes internal rate of return of series of up to 31 cash flows.

BONDS AND NOTES
PRICE Stores or computes price of bond or note.
YIELD Stores or computes yield (percentage) of a bond or note.
IS, ST Stores the issue and settlement dates of bond or note for calculations.
MT Stores the maturity date of a bond or note.
CALL Stores the call price or redemption value of a bond or note.
CPN Stores the coupon amount (percentage) for bond or note calculations.

DEPRECIATION
SL Calculates straight-line depreciation schedule.
SOYD Calculates sum-of-the-years digits depreciation schedule.
DB Calculates declining balance depreciation schedule.
BOOK Stores book value of an asset.
LIFE Stores depreciable life of an asset.
SAL Stores salvage value of an asset.
N1 Stores the starting year for a depreciation schedule.
N2 Stores the ending year for a depreciation schedule.

PERCENTAGE
% Computes percent.
Δ% Computes percent of change between two numbers.
%Σ Computes percent one number is of a total.

CALENDAR
2000 Year Calendar October 15, 1582 to November 25, 4046.
DATE + DAYS Computes a future or past date from a given date and a fixed number of days.
Δ DAYS Computes number of days between dates.
g PRINT x For a given date, prints its day of the week.

STATISTICS
Σ+ Automatically accumulates two variables for statistics problems: Σx, Σy, Σx², Σy², Σxy, and number of terms n.
Σ- Deletes statistical variables for changing or correction.
$\bar{x}$ Computes mean for x and y.
s Computes standard deviation for x, and y.
L.R. Linear regression of trend line.
$\hat{y}$ Linear estimate.
r Correlation coefficient.

STORAGE
STO Stores number in one of 30 storage registers. Performs storage register arithmetic upon 10 of the registers.
RCL Recalls number from one of 30 storage registers.

PRINTING AND CLEARING
AMORT Prints amortization schedule.
LIST: FINANCE Prints all values for compound interest problems, bonds and notes, and depreciation schedules.
PRINT x Prints contents of display.
LIST: STACK Prints contents of operational stack.
LIST: REG Σ Together print contents of 30 addressable storage registers.
CLx Clears display.
CL FIN Clears financial functions for new problem.
CL REG CL Σ Together clear 30 addressable storage registers.
CLEAR Clears entire calculator—display, operational stack, all storage registers, and financial functions.

NUMBER ENTRY AND MANIPULATION
ENTER↑ Separates numbers for arithmetic and other functions.
CHS Changes sign of displayed number of exponent.
x⇄y R↓ R↑ Functions to manipulate numbers in operational stack.
EEX Enter exponent of 10.
RND Rounds actual number in display to number seen in display.
LAST x Recalls number displayed before last operation back to display.

MATHEMATICS
y^x Raises number to power.
e^x Natural antilogarithm.
LN Natural logarithm.
$\sqrt{x}$ Square root.
1/x Reciprocal.
− + × ÷ Arithmetic functions.

PHYSICAL SPECIFICATIONS
WIDTH: 22.9 centimetres (9.0 in).
LENGTH: 20.3 centimetres (8.0 in).
HEIGHT: 6.35 centimetres (2.5 in).
WEIGHT: 1.13 kilograms (40 oz).
RECHARGER/AC ADAPTER WEIGHT: 170 grams (6 oz).
SHIPPING WEIGHT: 2.7 kilograms (5 lb 15 oz).

TEMPERATURE SPECIFICATIONS
OPERATING TEMPERATURE RANGE: 0° to 45°C (32°F to 113°F); with paper, 5% to 95% relative humidity.
CHARGING TEMPERATURE RANGE: 15° to 40°C (59° to 104°F).
STORAGE TEMPERATURE RANGE: −40° to +55°C (−40° to +131°F).

POWER SPECIFICATIONS
AC: Depending on recharger/ac adapter chosen, 115 or 230V +10%, 50 to 60 Hz.
BATTERY: 5.0 Vdc nickel-cadmium battery pack.
BATTERY OPERATING TIME: 3 to 7 hours.
BATTERY RECHARGING TIME: Calculator off, 7 to 10 hours; calculator on, 17 hours.

PRICE IN U.S.A.: $625.

MANUFACTURING DIVISION: CORVALLIS DIVISION
1000 N.E. Circle Boulevard
Corvallis, Oregon 97330 U.S.A.