Chapter 1
Introductory Mathematica syntax

Learn to use HELP and Documentation Center

Print a string

- Syntax: **Print[]**
- Every command line in Mathematica should by default end with a ";",

Print["hello world"];

- Assign a "value" to variable name:

stringname= "hello world";

- The "value" assigned to the variable stringname is not a numerical value, but a string.
- Syntax: **StringQ[]**
- Check that the variable stringname is assigned as a string using

StringQ[stringname];

Short cut keys

- Execute a code: Shift + Enter
- Quit kernel: Alt + V $\rightarrow$ Q $\rightarrow$ Enter
- Clear All Output: Alt + C $\rightarrow$ L $\rightarrow$ Enter

Print a value

- Assign a numerical value of 1 and 2 to the variable named a and b:

a=1;b=2;

- Define a new variable c as the sum of a and b:

c=a+b;

- Print out the value of c:

Print[c];

- Syntax: **NumberQ[]**
- You can check whether the variable c is a numeric using

NumberQ[c]

Do loop command

- Syntax:

Do [
expression;
,{n,nbegin, nlast, interval}
];

- Example:

Do [
Print[“hello world for the ”,n, ”time”];
,{n,1, 5, 1}
];

Semicolon “;”

- Each line of code in Mathematica must be ended with a semicolon “;”

e.g,

Print[“Hollo world”];

Use do loop to calculate the accumulative amount of money you owe to a long (charging you 2% per week) in 20 weeks.

```
amount=5000;interestrate=0.02;
Do[
interest=amount*interestrate;
amount= amount+interest;
Print[“This is the ”, n, “ week. The amount I owe a long is ”,
interest];
,{n,1,20,1}
];
```

Syntax: Input[], for interactive input

```
amount = Input["Amount borrowed"];
interestrate = Input["interest rate, in percentage"];
nlast = Input["how many month to clear the loan"];

Print["amount=", amount];
Print["interestrate=", interestrate];
Print["nlast=", nlast];
```

Improved version: replace 20 by nlast=20; use Input[] for interactive input

```
amount = Input["Amount borrowed"];
interestrate = Input["interest rate, in percentage"];
nlast = Input["how many month to clear the loan"];
Do[
interest=amount*interestrate;
amount= amount+interest;
Print[“This is the ”, n, “ week. The amount I owe a long is ”, interest];
,{n,1,nlast,1}
];
```

If[] and **Break[]** command

- Syntax: **If [condition, do something];**
- Syntax: **Break [];**

Exercise:

Modify the previous code so that you can calculate the amount you still owe along in each week, and how many weeks it takes to clear off the loan if you are paying an installment of *x* per week.

Incorporate the If and Break commands so that it can automatically break off the Do loop when the loan is cleared off.

Assume: amount borrowed=5000;rate=2% per week;amout pay back per month=200;

Sample code:
http://comsics.usm.my/tlyoon/teaching/ZCE111_1516SEM2/notes/mathematicafiles/C1_along1.nb

Robustness

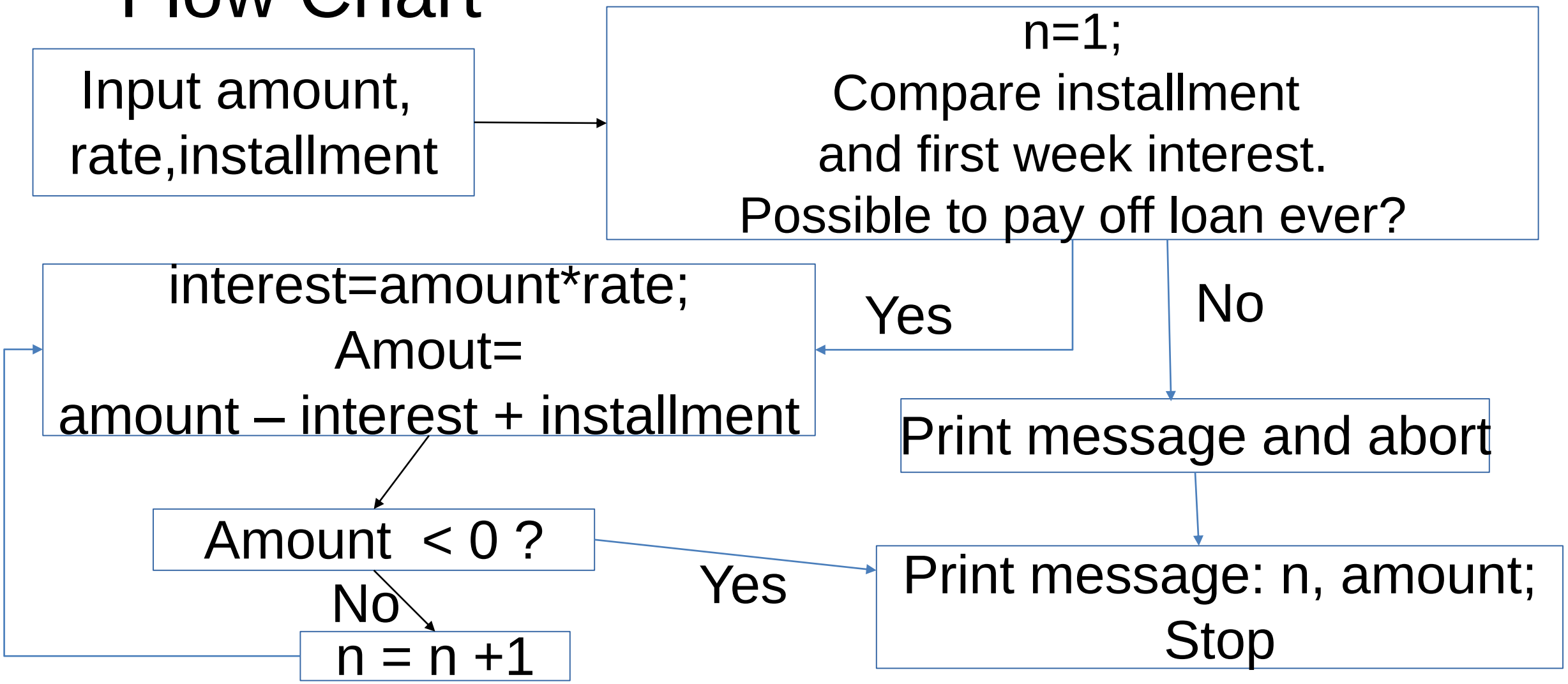
Your code should be ‘robust’, i.e.,

(i) It should be able to automatically abort the programe if your installment is insufficient to pay back the loan ever

(ii) The code must be able to provide the answer in which week the loan is cleared of (if ever), no matter how many weeks to takes, even if it is near infinity (this could occur if the installment is almost equal but just slightly larger than the first week interest).

If your code is robust enough, it could be used in the future without making any modification except reassigning the values to these variables.

Flow Chart


Input amount,
rate,installment
n=1;
Compare installment
and first week interest.
Possible to pay off loan ever?
Yes
No
interest=amount*rate;
Amout=
amount – interest + installment
Print message and abort
Amount < 0 ?
Yes
No
Print message: n, amount;
Stop
n = n +1

Syntax: **While[]**

The previous along code is better implemented using the syntax

While[ ];

While sample code

Sample code:

```
i = 0;
While[i < 10,
  i = i + 1;
  Print[{"i=", i}]
  ];
```

Sample code:

```
amount = 5000; rate = 2; installment = 500; n = 0;
While[amount > 0,
 n = n + 1;
amount = amount +amount* (rate/100.0) - installment;
 Print[{n, amount}];
 ];(*end while*)
```

Plotting amount owed vs. week

The aloong while loop code has all the loan information for every week, e.g., amount owe from week 0 till the week you pay off the loan.
So, can you plot the graph of amount owed to along as a function of number of week?

Syntax: **List, Table, ListPlot**

Sample code:

```
amount = 5000; rate = 0.02; installment = 500; n = 0;
While[amount > 0, n = n + 1;
 amount = amount + (rate/100.0)* amount - installment;
 dummy[n] = amount;
 Print[{n, dummy[n]}];
 ];(*end while *)
ldummy = Table[dummy[i], {i, 1, n - 1}]; (* ldummy  is a
“List” *)
lidummy = Table[{i, dummy[i]}, {i, 1, n - 1}] (* ldummy  is a
“List” *)
ListPlot[lidummy]
```

Customising **ListPlot**

```
Needs["PlotLegends`"];
lpt = ListPlot[lidummy,
   Joined -> True,
   Mesh -> All,
   PlotLabel -> "Amount owed vs. number of week",
   AxesLabel -> {"week", "amount owed"},
   PlotMarkers -> "\[Times]",
   PlotStyle -> {Red, Dotted, Thin},
   PlotLegend -> {"Amount owed"}
   ];
Print[lpt];
```

Summary

- In this Chapter, you have learned:
- Execute a code: Shift + Enter
- Quit kernel: Alt + V $\rightarrow$ Q $\rightarrow$ Enter
- Clear All Output: Alt + C $\rightarrow$ L $\rightarrow$ Enter
- **Print[]; Do[]; NumberQ; StringQ; If[]; Break[];**
- Flowchart
- **While[];**
- Assign a value to an element in a list: **dummy[n]** = value;
- **List; Table; ListPlot;**
- Customising a plot: **Joined,Mesh -> All,PlotLabel,**
  AxesLabel,PlotMarkers,PlotStyle,PlotLegend

Assignment

- With all these syntax, you must have already able to complete Assignment 1