

# ZCE 111

# Computational Approach in Physics Learning

Instructor

Yoon Tiem Leong

School of Physics

Universiti Sains Malaysia

11800 USM, Penang, Malaysia

# Linux compute nodes

- Log in to the Linux compute nodes using your username and password.
- We will refer to the computers (PC) in the Rocks Computer Cluster as “compute nodes”.

# Using Linux Terminal

- In your Linux computer, open a terminal
- Syntax: Use “top” to check out the processes in your compute node (local PC).
- Syntax: Use “hostname” to check out the name your compute node.
- Syntax: Use “whoami” to check out your username.
- Syntax: Use “pwd” to check out your current directory
- Syntax: Use “ls” to check out the names of files and folders in your current directory.

# Using Linux Terminal

- Syntax: Use "nautilus --browser" to launch the "file explorer".
- Syntax: Use "gedit" to launch the word editor "gedit".

# Launching a Mathematica session

- Launch Mathematica session in your compute node: either by typing "Mathematica" in the terminal, or click Start->System->Mathematica.
- If prompted for license authentication, choose to authentication via license server. These server's IP address are "comsics.usm.my" and "anicca.usm.my".
- Open a new notebook (\*.nb) file, save it as "hello.nb" in your current folder.
- Syntax: Check your current folder where "hello.nb" is saved using Directory[].
- Syntax: Check the files in your current folder using FileNames[].

# What to do if your compute node hangs?

- Remedy 1: Press ctrl+alt+delete
- Remedy 2: Log in to your node (say compute-0-10 in anicca) from another compute node using your current username. Type `ps -u username`, where *username* stand for your username (e.g., human2). Then kill off the process you think is causing the problem in compute-0-10 by typing `kill -9 XXXX`, where `XXXX` stands for the process id (e.g., 334320). Alternatively, kill the processes by typing `pkill -9 process`, where *process* stands for the name of the process you want to kill, e.g., Mathematica or MathKernel.
- Remedy 3: Log in to your node (say compute-0-10 in anicca) from another compute node using your current username. Then type `/usr/sbin/gdm-restart`
- Remedy 4: Yell "SOS".

# 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];`

# 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}  
    ];
```

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

```
amount=1000;interestrates=0.02;
```

```
Do[
```

```
interest=amount*interestrates;
```

```
amount= amount+interest;
```

```
Print["This is the ", n, " week. The amount I owe a long is ",  
interest];
```

```
,{n,1,20,1}
```

```
];
```

## Improved version: replace 20 by nlast=20;

```
amount=1000;interestrates=0.02;nlast=20;
```

```
Do[
```

```
interest=amount*interestrates;
```

```
amount= amount+interest;
```

```
Print["This is the ", n, " week. The amount I owe a long is ",  
interest];
```

```
,{n,1,nlast,1}
```

```
];
```

## Exercise:

- Modify the previous code so that you can calculate the amount you still owe a long every week if you are paying back him a fixed amount  $x$  every week.
- Problem 1: Use your code to determine how many weeks it takes to clear off the loan if you are paying an installment of  $x$  per week.
- Problem 2: Modify your code to determine what is the installment you need to pay every week if you wish to finish off the loan in the  $N$ th week.
- Note: 4 variables are involved here: initial amount borrowed, interest rate, weekly installment and total number of week to clear the loan.

# If and Break[] command

- Syntax: If [condition, do something];
- Syntax: Break [];
- Try to incorporate the two commands above in your code so that it can automatically break off the Do loop when the loan is cleared off. Without these, you will have to do a trial and error every time a new amount or new interest rate are introduced.

# Please complete Assignment 1