UTS
UNIVERSITY OF TECHNOLOGY SYDNEY

PROGRAMMING FUNDAMENTALS

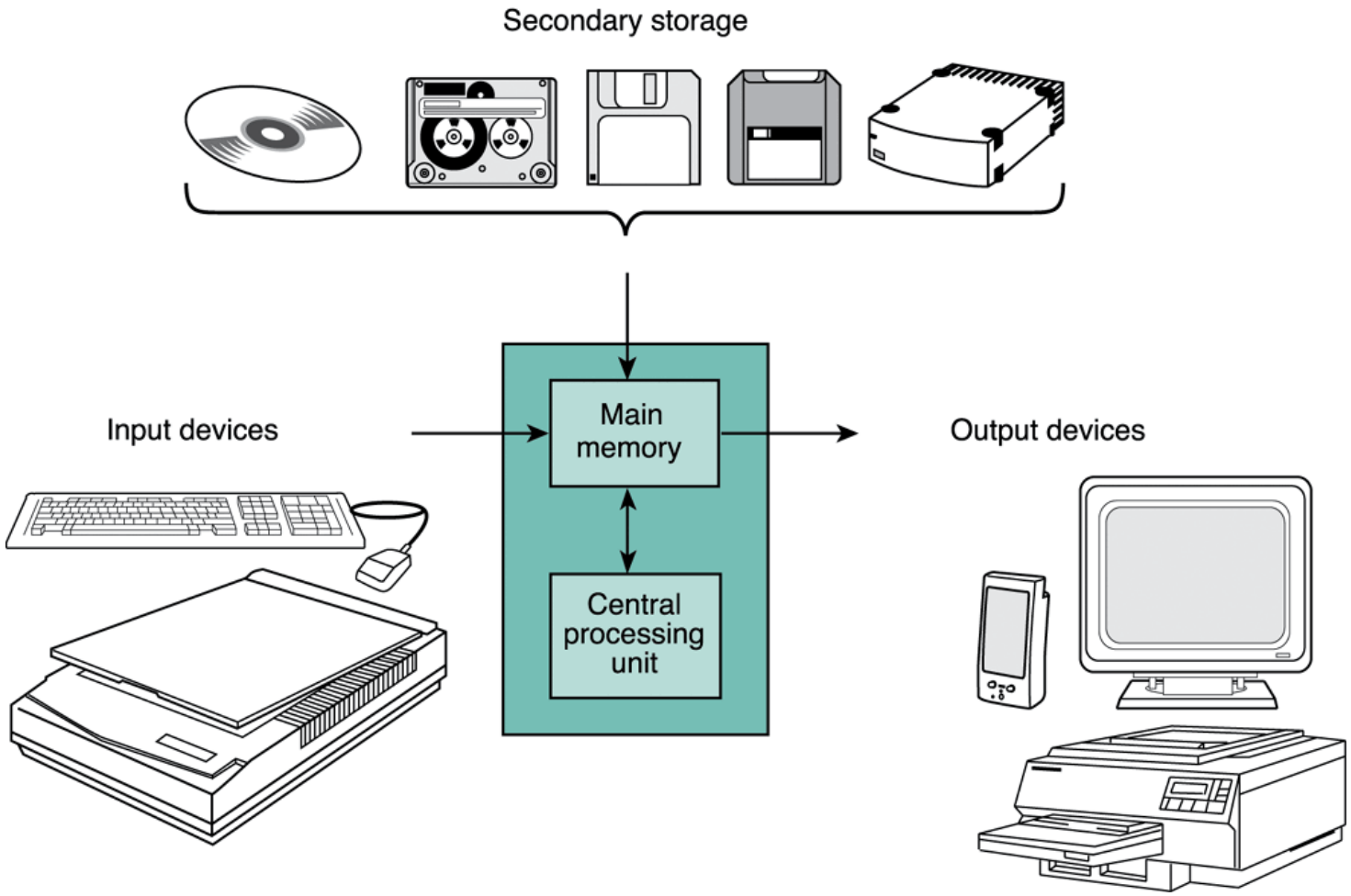
BEESHANGA ABEWARDANA JAYAWICKRAMA

UTS:
ENGINEERING AND
INFORMATION
TECHNOLOGY

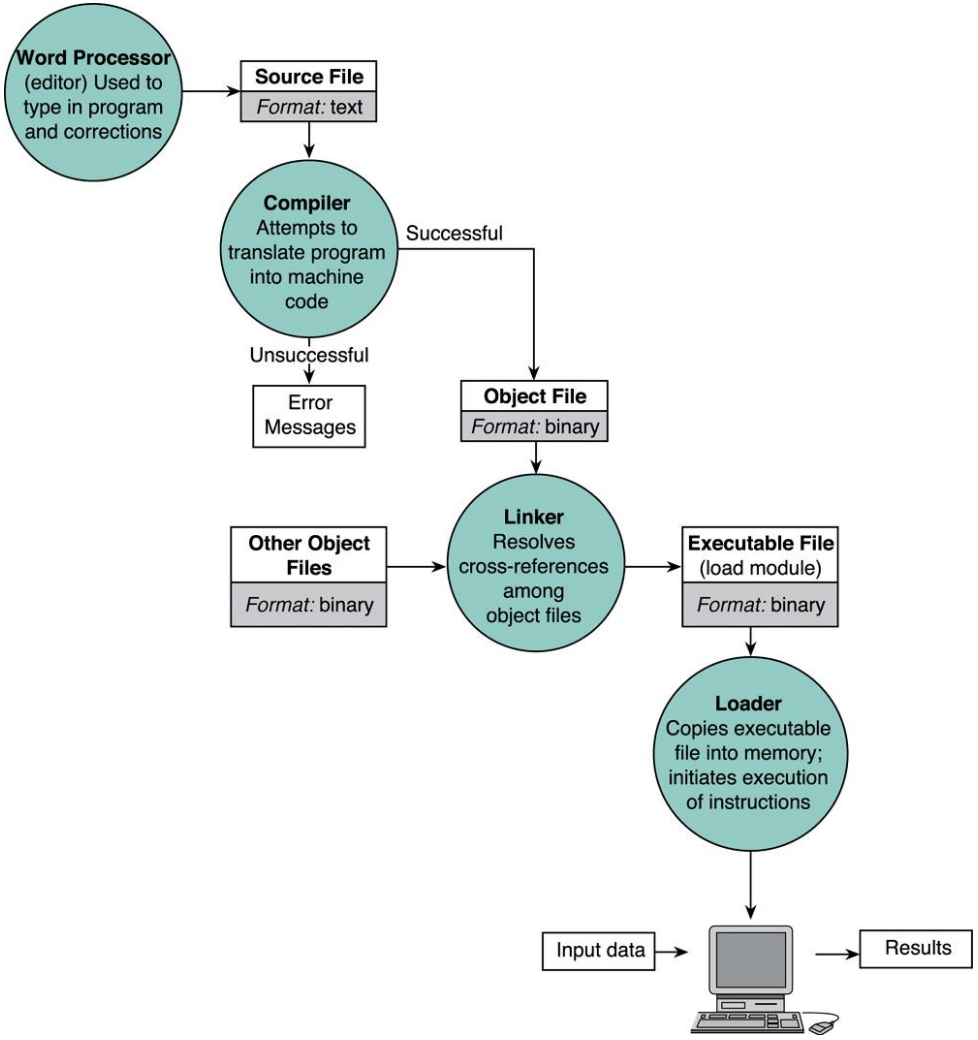
UTS CRICOS PROVIDER CODE: 00099F

feit.uts.edu.au

COMPONENTS OF A COMPUTER



RUNNING A HIGH LEVEL LANGUAGE PROGRAM



PREPARATION WORK…

```
gcc –Wall –Werror –ansi –o helloworld.out helloworld.c -lm
```

What do those terms mean?

What happens if you type:

```
gcc –Wall –Werror –ansi helloworld.c -lm
```

Why do we use:

`–Wall –Werror –ansi` and `-lm`

PREPARATION WORK…

```
#include <stdio.h>
int main(void){
        printf("Hello World\n");
        return 0;
}
```

1. what does stdio stand for? what does #include do?
2. what is main? could it be something other than "main"?
3. what is printf?
4. what is \n?
5. what is return 0?

SOFTWARE DEVELOPMENT STEPS

1. Problem requirement – understand what the program is expected to do
2. Problem analysis – identify the inputs, outputs and required resources
3. Solution design – design the algorithm
4. Implement – write the code
5. Test – think about possible test cases and thoroughly verify the program
6. Maintain and update

Note: first 3 steps are done on paper.

SOFTWARE DEVELOPMENT STEPS EXAMPLE

Write a program to convert a distance given in miles to kilometres.

1. Problem requirement – convert distance from miles to kilometres
2. Problem analysis

   input: miles – the distance in miles from the user, data type double.

   output: kms – the converted distance in kms, data type double.

   resources: 1 mi = 1.609 km
3. Solution design

   Get the distance in miles from the user

   Convert the distance to kms

   Display the kms on the screen
4. Implement – shown in the class
5. Test – shown in the class
6. Maintain and update – not covered in this subject

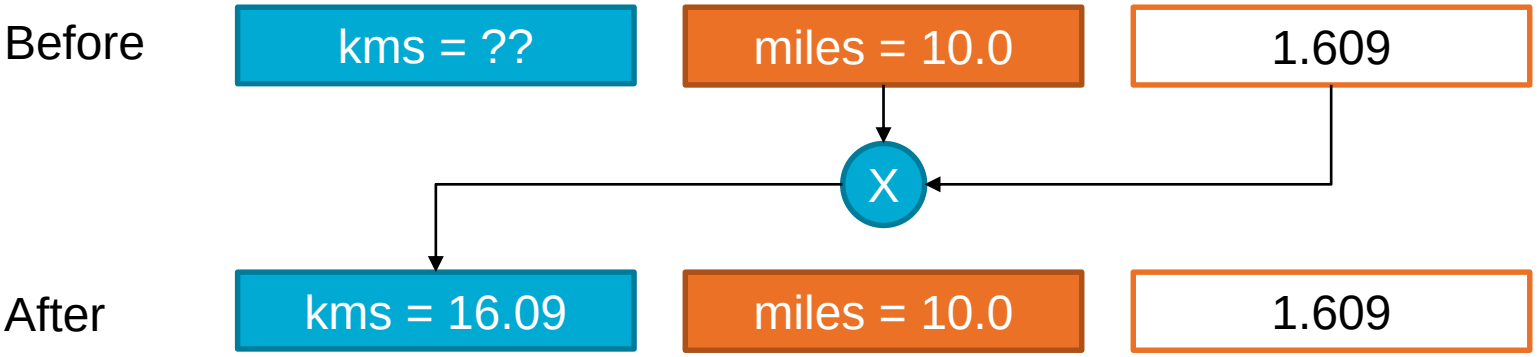
VARIABLES

There two stages after which variable value can be used:

> `double kms;` → definition/declaration- allocate memory to store a double. Provide variable **name** and the **type**.

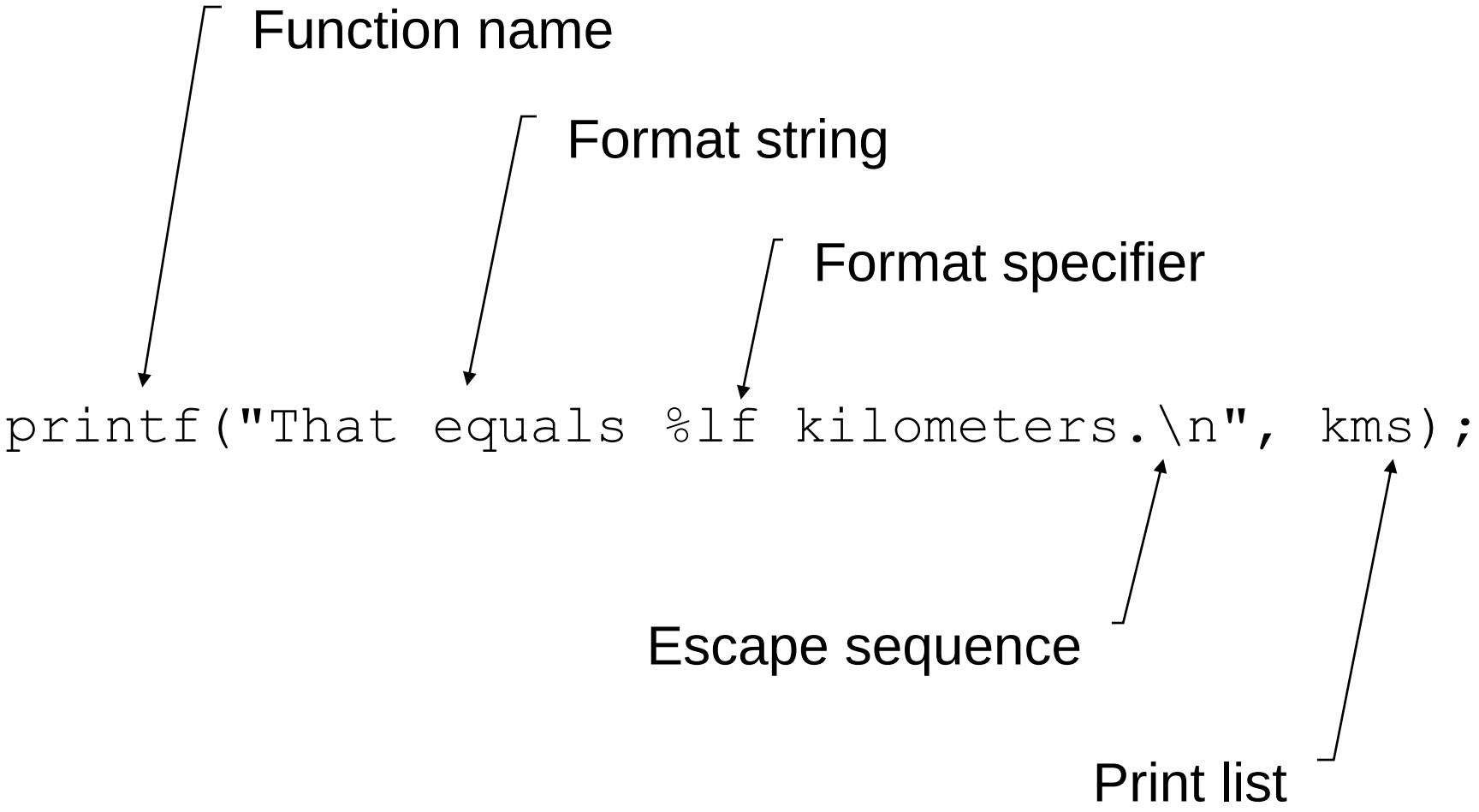
kms = ??

> `kms = KMS_PER_MILE * miles;` → initialisation- first time a value is assigned



Value of the variable is "garbage" until initialisation happens, hence should not be read.

PRINTF


Function name
Format string
Format specifier
printf("That equals %lf kilometers.\n", kms);
Escape sequence
Print list

SCANF

```
int first, second;

scanf("%d%d", &first, &second);
```

The `&` symbols tell the system to put the collected values into the memory locations of `first` and `second` – more about variable values and locations later in the subject.

`scanf` gets values from the user, so it must know the location in memory to put the values (not the values previously stored in memory).

In contrast, `printf` needs to know the values stored in memory to show on the screen (not the location in memory where values are stored).

TYPES OF ERRORS IN PROGRAMS

Compilation errors – using incorrect syntax. The compiler will pick these errors.

Logical errors – incorrect logic and operations. Only you can pick these errors through comprehensive testing.

Runtime errors – errors that happen in special corner cases. Only you can pick these errors through comprehensive testing.

GOOD PROGRAMMING HABITS

Programs written in this subject should

> Functionally accurate – hence testing is important.

> Follow good programming habits (more about this during the lab…)