**C for Everyone: Programming Fundamentals Notes**

**Week 1**

**Chapter: Example-Marathon**

* void in int main(void){} means there are no arguments.

**Chapter: Simple input/output – fahrenheit**

* %d in scanf(“%d”,&fahrenheit) means integer

celsius = (fahrenheit – 32)/1.8 . Even the two variables are integer. After divided by a float type, the product will also become float type**.**

**Chapter: Simple input/output – miles**

* A normal machine is 4 byte machine, int type is 4 bytes.

For a longer bytes int, we can use long long, which is 8 bytes.

**Code Practise**

**Week 1**

**Example-Circle code: Circle and Area**

**Example-Marathon: The distance of a marathon in kilometers**

**Simple input/output-fahrenheit: Conversion of Fahrenheit to Celsius**

**Simple input/output-miles: Distance of a marathon in yards**

**Assignment: Fix Dr. P’s mistake (week 1)**

**Week 2**

**Chapter: Comments**

* How the C compiler works?

First off, there is a preprocessor, which puts it in whatever code is necessary. There’s what’s called the tokenizer, and the tokenizer goes and looks through things and discard the comment.

* There are two comment styles. /\* \*/ and //

**Chapter: Keywords**

* Special words for the pre-processor like include may not be reserved as the keywords. Also, main is not a reserved/keyword, but a identifier.
* Search keywords in google by typing ANSI C/C++ keyword Table

**Chapter: Identifiers**

* Identifiers can be interpreted as the name of variables

**Chapter: Expressions and precedence (File included)**

* Precedence:

\*is higher precedent than binary + and binary –

But unary – and unary + (e.g. +8, and -8) has a higher precedence.

* Associativity:

a+b+c: from left to right -> ((a+b)+c)

a=b=c=3: from right to left -> (a=(b=(c=3)))

**Chapter: Expressions and evaluation**

c = ++a + b++; d += 5; //

Output: a = 6, b = 8 b = c = 5 + 7 + 1 = 13, d = -12 + 5 = -7

b will perform self-addition after the addition of c is done.

**Chapter: Fundamental types and sizeof**

* unsigned means rule out the negative numbers (only positive), this type is strictly be used in integers.
* long can be applied to both integers and to the doubles, which is used for large value number.
* 3 types of doubles: float, double, long double. Long double can store the largest value.

**Chapter: The integer and floating point types**

* Different representation of the same number:

%e or %E print as 1.23456e+00

%f print as 1.23456

%g or %G will represent like %e or %f depending on which one has the shortest representation on printing to the screen.

* float type should use %f in prinf()
* double and long float should use %lf in printf()

**Code Practise**

**Week 2**

**Expression and evaluation: Expression Evaluation**

**Declarations: Fundamental Types Declaration and Assignment**

**Fundamental types and sizeof: Fundamental Types sizeof operator**

**The char type: Char in c**

**The int type: Fundamental Types INT**

**Assignment: Fix Dr. P’s mistake (week 2)**