**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)**

**Week 3**

/\* C for Everyone: Programming Fundamentals

Week 3: Declaration \*/

**Chapter: Logical operators, expressions, and short-circuit evaluation**

* short-circuit evaluation:

If one part of conditions already determines the result, we can use else to represent other conditions. It can save a lot of computations.

For example:

if (outside && weather)

printf("\nPlease use an umbrella.\n");

else

printf("\nDress normally.\n\n");

There is only one truth result, so we can use short-circuit evaluation.

* The comparison and relational operator such as >, <, && has a very low precedence. For example, (a<3) && (a>5) will first compare a and 3 and a and 5. Then, operates the && conditions and returns a 0.
* if(!a) equals if(a == 0)

**Chapter: The conditional statement if and if-else**

* if statement is done by determining the expression equals zero or non-zero value. For example: if (speed < 65). When speed > 65, expression is false and evaluates to 0. Otherwise, it evaluates to 1.

**Chapter: The iterative statement while**

* The while statement:

while(expression):

statement

If expression is false (0), we omit the statement.

If expression is true (1), we execute the statement and repeat until the expression is false.

* Syntax

e1;

while (e2)

{

statement;

e3;

}

In typical case: e1 is Initializer

e2 is terminating condition

e3 is increment/ decrement

**Chapter: The for statement and its while analog**

* The for statement:

for (i = 1, i <= 5, i++) :

statement

* Syntax

for(e1, e2, e3):

statement

In typical case: e1 is Initializer

e2 is terminating condition

e3 is increment/ decrement

**Chapter: for statement code example**

* The initializer is not necessary. For example:

for (; (c = getchar()) != EOF; total\_chars++)

There is no initializer, as the terminating condition gives sufficient information to know when should the loop be ended. ctrl+c to exit the program to reach EOF.

**Chapter: oddball operators-conditional and comma**

* c = (a < b)? a:b means if (a < b) then c = a; else c = b;
* Syntax

expression 1?expression 2:expression3

**Chapter: ternary-operator code example**

* In the code practice

speed = (speed <= 65) ? (65) : (speed <= 70) ? (70) : (90);

can be split into two sessions.

(speed <= 65) ? (65) : (speed <= 70) ? (70) : (90)

Means if speed <= 65 then speed = 65, else = the speed is determined by the second sessions.

Then (speed <= 70) ? (70) : (90);

means if speed <=70 then speed = 70, else = 90

**Chapter: Break and continue and switch**

* Example of switch:

switch (i)

{

case 1: a = 2 \* i; break;

case 2: a = 2 \* i; break;

default: a = 4 \* i; break;

}

break; is to prevent executing the next case.

* The labels in the switch can only be integer types like char, int, long.

**Code Practise**

**Week 3**

**Logical operators, expressions, and short-circuit evaluation:**

logical operators and short circuit evaluation

**Logic operators quiz**

**The conditional statement if and if-else:** if and relations and flow of control

**while-cnt-char explained:** while-cnt-char-explained

**while-code-example:** Love you how much

**While loop questions**

**for statement code example:** demonstrate loop with for statement

**ternary-operator code example:** ternary operator and flow of control

**Switch questions**

**Assignment: Fix Dr. P’s mistake (week 3)**

**Cond-comma-ops quiz**