CmpE 150 - Week 2

Section - 02

Teaching Codes Account Info Will Be Sent

- Received your credentials for Teaching Codes system?
- https://programming.cmpe.boun.edu.tr

Recap

Last week:

- Introduction to C
- How to setup Eclipse
- Eclipse Introduction
- Writing “Hello World”
- Escape characters
- Special Keywords

https://github.com/melsener/cmpe150

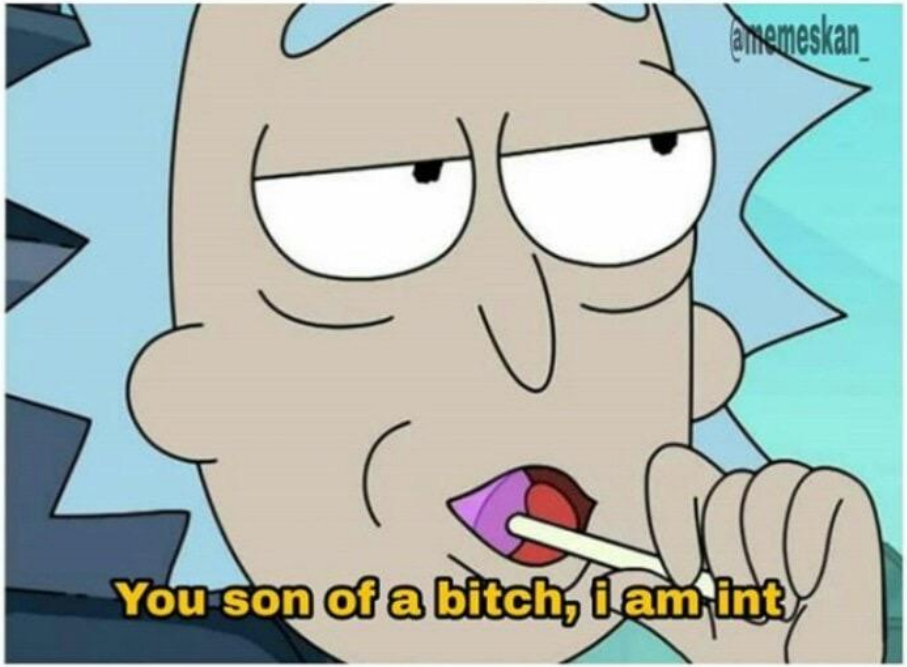
Arithmetic Operators

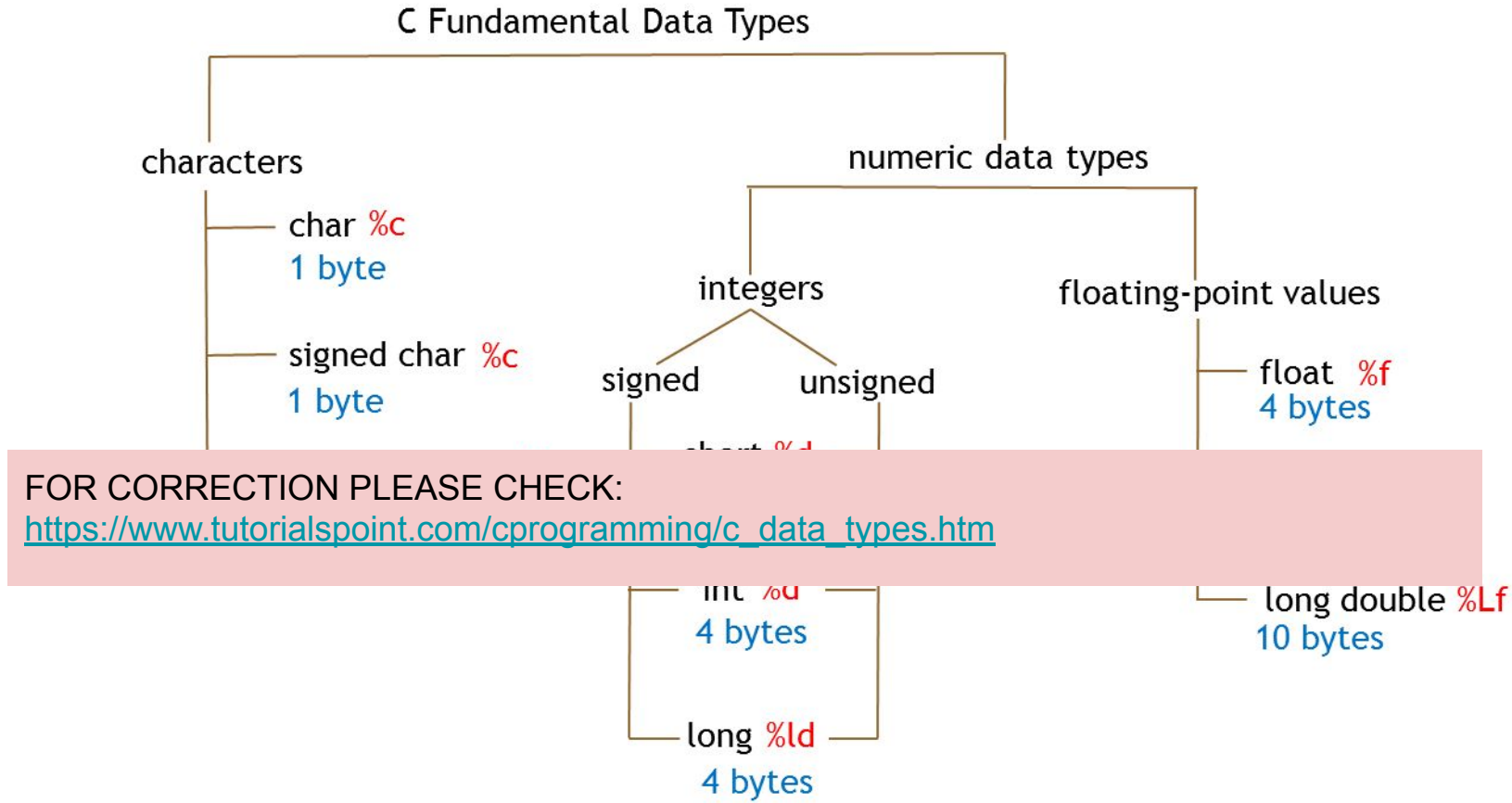
C operation	Arithmetic operator	Algebraic expression	C expression
Addition	+	$f + 7$	`f + 7`
Subtraction	-	$p - c$	`p - c`
Multiplication	*	bm	`b * m`
Division	/	x / y or $\frac{x}{y}$ or $x \div y$	`x / y`
Remainder	%	$r \bmod s$	`r % s`

Fig. 2.9 | Arithmetic operators.

```
int a;
a=6.9;
```

Data type-





FOR CORRECTION PLEASE CHECK:
https://www.tutorialspoint.com/cprogramming/c_data_types.htm

Src:
https://github.com/hkmztrk/CMPE150/blob/master/W02/w02_datatypes.png

scanf function

```
int x;
char y;
float z;
double t;
```

Format Specifier	Data Type	Example
%d	Decimal Integer	scanf(“%d”,&x);
%c	Char	scanf(“%c”,&y);
%f	Float	scanf(“%f”,&z);
%lf	Double	scanf(“%lf”,&t);

& is an address operator. For that moment, you must just know that it must be used before the name of a variable in the scanf() function.

Special Keywords

Keywords			
auto	double	int	struct
break	else	long	switch
case	enum	register	typedef
char	extern	return	union
const	float	short	unsigned
continue	for	signed	void
default	goto	sizeof	volatile
do	if	static	while

Keywords added in C99 standard

`_Bool` `_Complex` `_Imaginary` `inline` `restrict`

Keywords added in C11 draft standard

`_Alignas` `_Alignof` `_Atomic` `_Generic` `_Noreturn` `_Static_assert` `_Thread_local`

Fig. 2.15 | C's keywords.

Src: Deitel C How to Program

Casting

Converting values of one type to another type. Can be:

- **Explicit Casting (we'll talk about this)**
- Implicit Casting

How to do explicit casting?

```
(type_name) expression
```

Source: Tutorialspoint -Type Casting

Limits - integer vs long

```
#include <stdio.h>
#include <limits.h>

int main() {

   printf("The number of bits in a byte %d\n", CHAR_BIT);
   printf("The minimum value of SIGNED CHAR = %d\n", SCHAR_MIN);
   printf("The maximum value of SIGNED CHAR = %d\n", SCHAR_MAX);
   printf("The maximum value of UNSIGNED CHAR = %d\n", UCHAR_MAX);
   printf("The minimum value of SHORT INT = %d\n", SHRT_MIN);
   printf("The maximum value of SHORT INT = %d\n", SHRT_MAX);
   printf("The minimum value of INT = %d\n", INT_MIN);
   printf("The maximum value of INT = %d\n", INT_MAX);
   printf("The minimum value of CHAR = %d\n", CHAR_MIN);
   printf("The maximum value of CHAR = %d\n", CHAR_MAX);
   printf("The minimum value of LONG = %ld\n", LONG_MIN);
   printf("The maximum value of LONG = %ld\n", LONG_MAX);
   return(0);
}
```

Code Source: [Tutorialspoint - C Library <limits.h>]