COP3502 Module 03 Review



**Methods and Functions:**

Function- a named block of instructions

Method- a named block on instructions inside of a class

Method Call- causes method to execute

Parameters

Graphical user interface, text

Description automatically generated

Output: Hello, I’m Bat Man.

Method **Call**

Method

Argument

A **method** is like Bob. When you call “Bob” (**method call**) and tell him to build you a bird house he will (**method executes instructions**). First though, Bob tells you he needs a steel hammer, and birch wood (**parameters** **with their datatypes**). You give him a hammer and wood like he asks for (**arguments**). Bob builds the birdhouse!

Method Overloading- having more than one method with the SAME NAME, but different parameters.

Ex] public static void sayHello ( ){…}

public static void sayHello (String name){…}

Scope:

A variable declared inside of a function can only be used by THAT function. THAT function is the variables **scope**.

The **scope** of a variable is limited to the block it is declared in – sometimes a loop

Return value- the result of the method executing 🡪 what the method “spits out”

Ex] **public static int addTwoNumbers(int a, int b)**

**{**

**return a + b;**

**}**

**\*\*\*NOTE\*\*\*** Methods make **COPIES** of arguments – they do not use the actual variable passed in



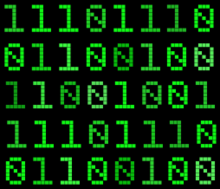
Return type

Method name

Parameters

Access level

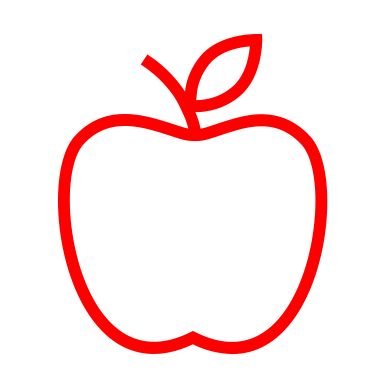
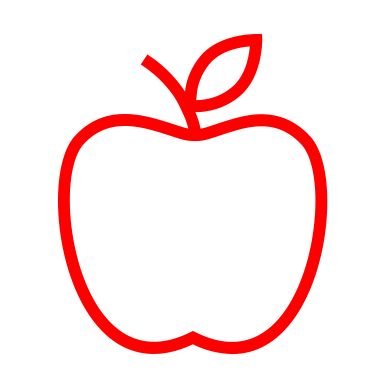
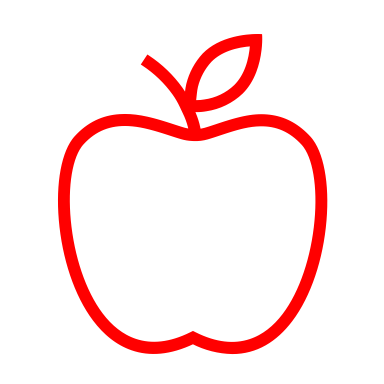
COP3502 Module 03 Review





**Number systems (Systems of counting):**

Binary- number system (system of counting) used by computers

There are lots of ways to count… you can count by using the tally system... = 

You can count by using the decimal system… **0 1 2 3 4 5 6 . . .**

Base **10**

And you can count using the **binary** system… **0 01 10 011 100 0101 0110 . . .**

Base **2**

So, if computers count in binary and we (humans) count in decimal (usually) how can we communicate with our computer??? **🡪 WE CAN CONVERT BINARY TO DECIMAL AND DECIMAL TO BINARY ☺**

***How to convert…***

Decimal 🡪 Binary:

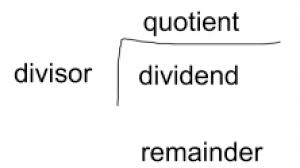
1] Divide decimal number by **2**. You will get a QUOTIENT and a REMAINDER

2] write down REMAINDER to keep track of it

3] Repeat steps 1 and 2 (using RESULT as the new decimal number) until your RESULT is 0.

4] list of REMAINDERs backwards is your binary answer ☺

Ex] decimal number = 6



Remainders:

0

1

1

6/2 🡪 QUOTIENT = 3 REMAINDER = 0

3/2 🡪 QUOTIENT = 1 REMAINDER = 1

1/2 🡪 QUOTIENT = 0 REMAINDER = 1

So 6 in binary is 110 which is equivalent to 0110

Binary 🡪 Decimal:

1] Line up all of your binary digits

2] Starting from the left and moving to the right, write increasing powers of **2** under each binary digit

3] Find all the “1” digits in your list. Multiply the 1 by the power of 2 written under it.

4] Add all of your multiplications together

Ex] binary number = 0110

0 1 1 0

2^3 = 8 2^2 = 4 2^1 = 2 2^0 = 1

1 \* 4 1 \* 2

Octal and Hexadecimal are two other number systems that we can use (systems of counting)

Octal has Base **8**

Hexidecimal has Base **16**

4 + 2 = 6

So, 0110 in binary is 6 in decimal