CSC111 – Jan 27th

Admin

- Jupyter Lab maintenance
  - Friday Jan 28th 8am - 10am
- Assignment submissions
  - Use filename exactly as specified.
  - **Assignment1.c (2)** is not acceptable
  - **assignment1.c** was specified
- UVIC emails – please use when contacting me.
  - https://webmail.uvic.ca

Review: A conditional statements can be inside a conditional statement

```
if( boolean_expression_1 ){
    if( boolean_expression_2 ){
        statement1;
        statement2;
    } else {
        statement3;
        statement4;
    }

    ...
}
```

Example

```
void commute(int rain _forecast, int wind_ forecast){

   if(rain_forecast > 10){
      if(wind_forcast > 20){
         printf("Take the bus\n");
      } else {
         printf("Ride your bike\n");
      }

      printf("Remember your raincoat\n");

   }else{
      printf("Ride your bike. It is nice!\n");
   }
}
```

Demo

Short circuit evaluation

Logical Operator	Example Boolean expression	Result of Boolean expression
`||`	`x < y || y < z`	The x<y evaluates first -> If it is True, the rest of the expression is not evaluated If it is False, the y<z is evaluated, then finally the **||** expression is evaluated
`&&`	`x < y && y < z`	The x<y evaluates first -> If it is False, the rest of the expression is not evaluated If it is True, the y<z is evaluated, then finally the **&&** expression is evaluated

Demo

Function Return Values

Defining a function with a return value

General form:

Any valid C type: `int`, `double`, `float`…

Function prototype

```
type fn_name(type param_name);
```

Must match

```
type fn_name(int param_name){
    C statement 1;
    C statement 2;
    return x;
}
```

types must match

Function definition

Concrete example:

```
int add1(int num);
```

```
int add1(int num) {
    int return_value;
    return_value = num + 1;
    return return_value;
}
```

Documentation expands…

```
/* Purpose: calculates num + 1
 * Parameters: int num – a number
 * Returns: int – the value of num + 1
 */
int add1 (int num) {
    int return_value;
    return_value = num + 1;
    return return_value;
}
```

- Every function you write SHOULD have a purpose comment
- If the function takes parameter(s), you MUST list them and the purpose must describe how the parameter(s) is(are) used.
- **If the function returns a value**, you must state the **type of the return** and a **description** of what is returned

```
#include <stdio.h>

int add1(int num);

int main ( ) {
    int x = 12;
    int result;
    result = add1(x);
    result = add1(11);
    return 0;
}

/* Purpose: calculates num + 1
 * Parameters: int num – a number
 * Returns: int – the value of num + 1
 */
int add1 (int num) {
    int return_value;
    return_value = num + 1;
    return return_value;
}
```

Function PROTOTYPE

Function CALLs passing expected **argument** **storing** the returned result

Documentation

Function DEFINITION

Demo