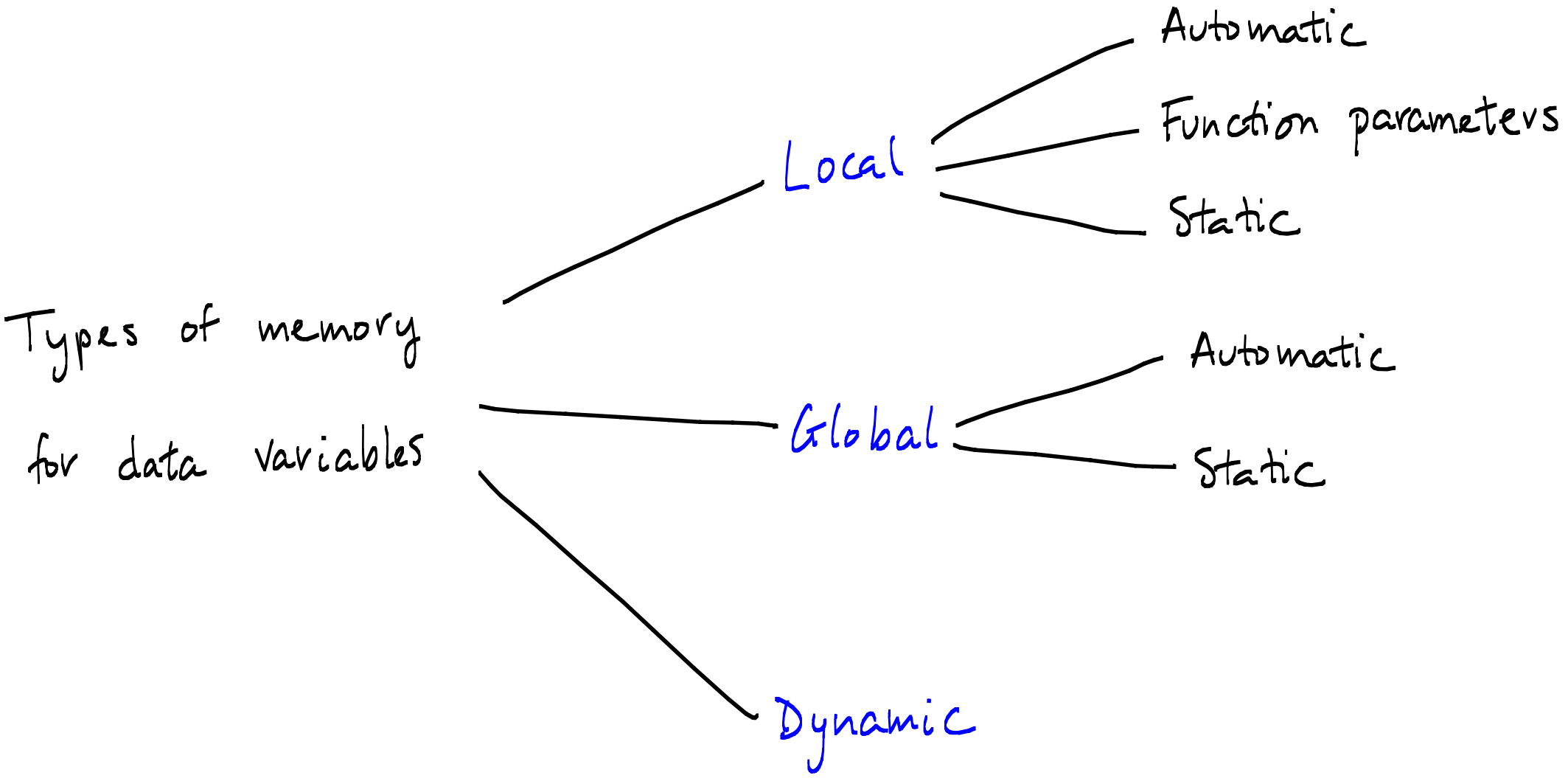
Introduction to Computing – B142L

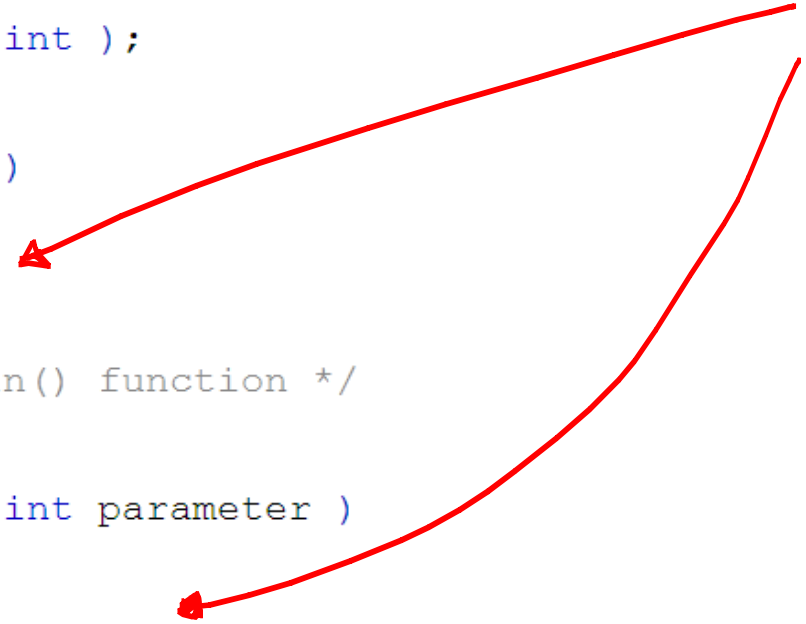
http://cnfolio.com/IndexIntroToComputing

1. **Getting started with C programming**
2. **Data types**
3. **Numbers**
4. **Control structures**
5. **Standard input and output**
6. **Arrays**
7. **Functions**
8. **Memory allocation**
9. **Preprocessor directives**
10. **Enums and structs**

December 7, 2009

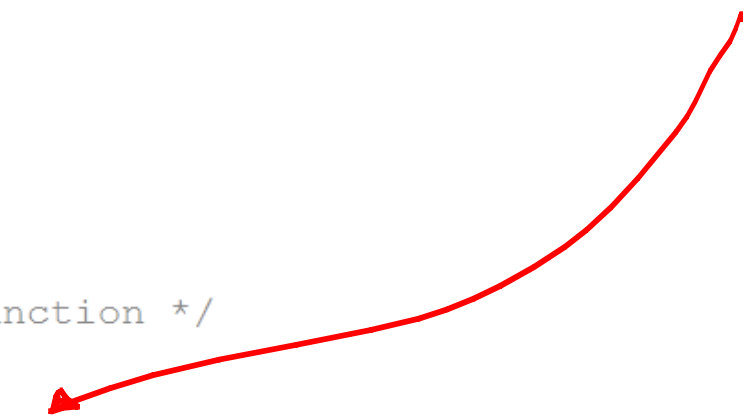

Types of memory for data variables
Local
Automatic
Function parametevs
Static
Global
Automatic
Static
Dynamic

```c
#include <stdio.h>


void function( int );


int main( void )
{


} /* end of main() function */


void function( int parameter )
{


} /* end of function definition */
```



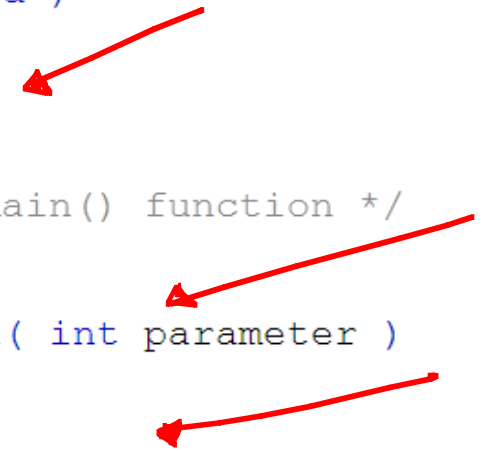
Local variables are declared inside a function.

```c
#include <stdio.h>


void function( int );


int main( void )
{


} /* end of main() function */


void function( int parameter )
{


} /* end of function definition */
```



Function parameters behave the same as local variables.

```c
#include <stdio.h>

void function( int );

int main( void )
{


} /* end of main() function */

void function( int parameter )
{

} /* end of function definition */
```



Local variables are:

1. Only accessible within the function where they were declared.
2. Re-initialized each time the function is called.

```c
void shuffle( int );


int main( void )
{
   int card10, card11, card20;
   card10 = 50;
   shuffle( card10 );
}


void shuffle( int card20 )
{
   int card30, card31;
   printf( "%d", card20 );
}
```

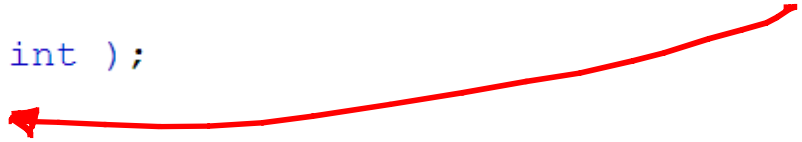
```c
#include <stdio.h>

void function( int );

int main( void )
{


} /* end of main() function */

void function( int parameter )
{


} /* end of function definition */
```

Static local variables are:

1. Only accessible within the function where they were declared.
2. Initialized once at the start of the program and keeps its value during the whole program.

```c
void shuffle( int );


int main( void )
{
   int card10, card11, card20;
   shuffle( card10 );
   shuffle( card10 );
   shuffle( card10 );
   shuffle( card10 );
}


void shuffle( int card20 )
{
   static int card30 = 600;
   printf( "%d\n", ++card30 );
}
```

```c
#include <stdio.h>


void function( int );


int main( void )
{


} /* end of main() function */

void function( int parameter )
{

} /* end of function definition */
```



Global variables are declared outside of all functions.

Typically defined at the top of the source code.

```c
#include <stdio.h>


void function( int );


int main( void )
{


} /* end of main() function */

void function( int parameter )
{

} /* end of function definition */
```

Global variables are:

1. Initialized once at the start of the program and keeps its value during the whole program.
2. Accessible anywhere in a program.

```c
void shuffle( void );

int cards[] = { 3, 6, 9, 12, 4, 8, 15 };

int main( void )
{
   shuffle();
   cards[ 2 ] = 7;
   shuffle();
   shuffle();
   shuffle();
   shuffle();
}


void shuffle( void )
{
   static int index = 0;
   printf( "%d\n", cards[ index++ ] );
}
```

```c
#include <stdio.h>


void function( int );


int main( void )
{


} /* end of main() function */

void function( int parameter )
{

} /* end of function definition */
```

Static global variables are:

1. Initialized once at the start of the program and keeps its value during the whole program.
2. Accessible anywhere in that source code file.

```c
#include <stdio.h>

#define MIN_EXAMPLE    4
#define MAX_EXAMPLE    20


void function( int );


int main( void )
{



} /* end of main() function */

void function( int parameter )
{

} /* end of function definition */
```

Constants defined using preprocessor commands are not variables!