COP 3223H: Introduction to C Programming

Fall 2023

University of Central Florida

Dr. Kevin Moran

Week 3- Class 11: User Defined Functions II

Today's Agenda

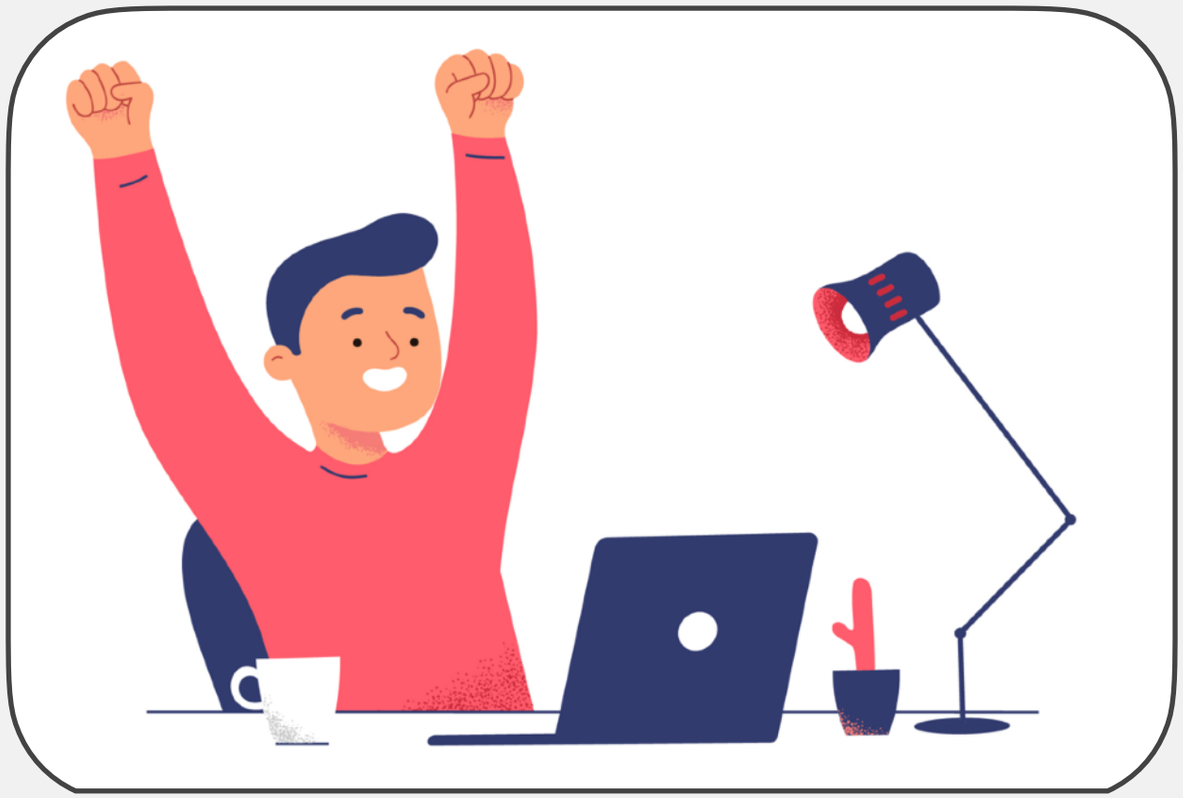
1. Review User-defined Functions
2. Discuss Functions with Arguments/Parameters

Introducing Functions!

- All C programs execute instructions in functions.
- All the programs we've looked at only had 1 function (`main`).
- Programs can get bigger based on the problem trying to be solved.
- It would be very convoluted with all instructions executed in the main function (*this is bad practice!*).
- In this lecture, we will learn that programmers can define their own user-defined functions to perform tasks.

Benefits of Defining our Own Functions

- Improves Readability
- Improves Reusability
- Helps to organize the abstractions in your program



User-defined Functions

- One way that programmers implement top-down design is defining their own functions (user defined functions)
- User defined functions are sets of instructions that are ***defined*** by the programmer
- Programmers will break down a larger problem into subproblems and will solve these subproblems in user-defined functions
- In order to invoke the function, you must ***call*** it

Types of Functions

- There are ***<u>two types</u>*** of functions.
  - Functions that *return a value*.
  - Functions that *don't return a value*.
- These types of functions are defined through their prototypes.
  - Functions that don't return a value have the reserved word **`void`** in front of the name.
  - Functions that do return a value have the type of data (**`int`**, **`double`**, **`char`**) in front of the of the name.

Functions with Arguments/Parameters

- Subproblems may require additional information in order to solve.
- Arguments send information over to functions.
- Input arguments are used to pass information into a function subprogram.
- Functions can have multiple arguments.

Functions with Arguments/Parameters

```
#include<stdio.h>

int mySecretForumla (int num, int num2, int num3);
int main ()
{
     int num1 = 3;
     int num2 = 2;
     int num3 = 1;

int x = mySecretForumla (num1,num2, num3) ;

printf ("x = 8d\n", x);

return 0;

}

int mySecretForumla (int num1, int num2, int num3)
{
int result = num1 + num2 * num3 - num3;
return result;
}
```

Parameters

Arguments

Pass By Value

- Whenever a function with arguments is called, they must share the values properly.
- One way of doing this is pass by value.
- Pass by value is when a value stored in memory (stack space) is ***copied*** and sent over to the proper parameter of the respective function (which is also stored in a different location of the stack space).
- The following set of slides shows a demonstration.

Pass By Value Example

Here

```c
#include<stdio.h>

void myFunction (int numl, int num2, int num3);

int main()
{
int num1 = 3;
int num2 = 2;
int num3 = 1;
printf (“num1 = %d\n", num1);
printf ("num2 = %d\n", num2);
printf ("num3 = %d\n", num3);

myFunction (num1, num2, num3);

printf ("num1 = %d\n", num1);
printf (“num2 = %d\n", num2) ;
printf ("num3 = %d\n", num3);
return 0;
}


void myFunction (int num1, int num2, int num3)
{
num1 = 5;
num2 = 8;

printf ("num1 = %d\n", num1);
printf ("num2 = %d\n" , num2);
printf ("num3 = %d\n", num3);
}
```

Stack Space	
AA9	
AA8	
AA7	
AA6	
AA5	
AA4	
AA3	
AA2	
AA1	
AA0	

Pass By Value Example

Here

```
#include<stdio.h>

void myFunction (int numl, int num2, int num3);

int main()
{
int num1 = 3;
int num2 = 2;
int num3 = 1;
printf (“num1 = %d\n", num1);
printf ("num2 = %d\n", num2);
printf ("num3 = %d\n", num3);

myFunction (num1, num2, num3);

printf ("num1 = %d\n", num1);
printf (“num2 = %d\n", num2) ;
printf ("num3 = %d\n", num3);
return 0;
}


void myFunction (int num1, int num2, int num3)
{
num1 = 5;
num2 = 8;

printf ("num1 = %d\n", num1);
printf ("num2 = %d\n" , num2);
printf ("num3 = %d\n", num3);
}
```

Stack Space	
AA9	
AA8	
AA7	
AA6	
AA5	
AA4	
AA3	
AA2	
AA1	
AA0	num1 = 3

Pass By Value Example

Here

```
#include<stdio.h>

void myFunction (int numl, int num2, int num3);

int main()
{
int num1 = 3;
int num2 = 2;
int num3 = 1;
printf (“num1 = %d\n", num1);
printf ("num2 = %d\n", num2);
printf ("num3 = %d\n", num3);

myFunction (num1, num2, num3);

printf ("num1 = %d\n", num1);
printf (“num2 = %d\n", num2) ;
printf ("num3 = %d\n", num3);
return 0;
}


void myFunction (int num1, int num2, int num3)
{
num1 = 5;
num2 = 8;

printf ("num1 = %d\n", num1);
printf ("num2 = %d\n" , num2);
printf ("num3 = %d\n", num3);
}
```

Stack Space	
AA9	
AA8	
AA7	
AA6	
AA5	
AA4	
AA3	
AA2	
AA1	num2 =2
AA0	num1 = 3

Pass By Value Example

Here

```
#include<stdio.h>

void myFunction (int numl, int num2, int num3);

int main()
{
int num1 = 3;
int num2 = 2;
int num3 = 1;
printf (“num1 = %d\n", num1);
printf ("num2 = %d\n", num2);
printf ("num3 = %d\n", num3);

myFunction (num1, num2, num3);

printf ("num1 = %d\n", num1);
printf (“num2 = %d\n", num2) ;
printf ("num3 = %d\n", num3);
return 0;
}


void myFunction (int num1, int num2, int num3)
{
num1 = 5;
num2 = 8;

printf ("num1 = %d\n", num1);
printf ("num2 = %d\n" , num2);
printf ("num3 = %d\n", num3);
}
```

Stack Space	
AA9	
AA8	
AA7	
AA6	
AA5	
AA4	
AA3	
AA2	num3 = 1
AA1	num2 = 2
AA0	num1 = 3

Pass By Value Example

Here

```
#include<stdio.h>

void myFunction (int numl, int num2, int num3);

int main()
{
int num1 = 3;
int num2 = 2;
int num3 = 1;
printf (“num1 = %d\n", num1);
printf ("num2 = %d\n", num2);
printf ("num3 = %d\n", num3);

myFunction (num1, num2, num3);

printf ("num1 = %d\n", num1);
printf (“num2 = %d\n", num2) ;
printf ("num3 = %d\n", num3);
return 0;
}


void myFunction (int num1, int num2, int num3)
{
num1 = 5;
num2 = 8;

printf ("num1 = %d\n", num1);
printf ("num2 = %d\n" , num2);
printf ("num3 = %d\n", num3);
}
```

Stack Space	
AA9	
AA8	
AA7	
AA6	
AA5	
AA4	
AA3	
AA2	num3 = 1
AA1	num2 = 2
AA0	num1 = 3

Pass By Value Example

Here

```
#include<stdio.h>

void myFunction (int numl, int num2, int num3);

int main()
{
int num1 = 3;
int num2 = 2;
int num3 = 1;
printf (“num1 = %d\n", num1);
printf ("num2 = %d\n", num2);
printf ("num3 = %d\n", num3);

myFunction (num1, num2, num3);

printf ("num1 = %d\n", num1);
printf (“num2 = %d\n", num2) ;
printf ("num3 = %d\n", num3);
return 0;
}


void myFunction (int num1, int num2, int num3)
{
num1 = 5;
num2 = 8;

printf ("num1 = %d\n", num1);
printf ("num2 = %d\n" , num2);
printf ("num3 = %d\n", num3);
}
```

Stack Space	
AA9	
AA8	
AA7	
AA6	
AA5	
AA4	
AA3	
AA2	num3 = 1
AA1	num2 = 2
AA0	num1 = 3

Pass By Value Example

Here

```
#include<stdio.h>

void myFunction (int numl, int num2, int num3);

int main()
{
int num1 = 3;
int num2 = 2;
int num3 = 1;
printf (“num1 = %d\n", num1);
printf ("num2 = %d\n", num2);
printf ("num3 = %d\n", num3);

myFunction (num1, num2, num3);

printf ("num1 = %d\n", num1);
printf (“num2 = %d\n", num2) ;
printf ("num3 = %d\n", num3);
return 0;
}


void myFunction (int num1, int num2, int num3)
{
num1 = 5;
num2 = 8;

printf ("num1 = %d\n", num1);
printf ("num2 = %d\n" , num2);
printf ("num3 = %d\n", num3);
}
```

Stack Space	
AA9	
AA8	
AA7	
AA6	
AA5	
AA4	
AA3	
AA2	num3 = 1
AA1	num2 = 2
AA0	num1 = 3

Pass By Value Example

```
#include<stdio.h>

void myFunction (int numl, int num2, int num3);

int main()
{
int num1 = 3;
int num2 = 2;
int num3 = 1;
printf (“num1 = %d\n", num1);
printf ("num2 = %d\n", num2);
printf ("num3 = %d\n", num3);

myFunction (num1, num2, num3);

printf ("num1 = %d\n", num1);
printf (“num2 = %d\n", num2) ;
printf ("num3 = %d\n", num3);
return 0;
}


void myFunction (int num1, int num2, int num3)
{
num1 = 5;
num2 = 8;

printf ("num1 = %d\n", num1);
printf ("num2 = %d\n" , num2);
printf ("num3 = %d\n", num3);
}
```

Here

Stack Space	
AA9	
AA8	
AA7	
AA6	
AA5	
AA4	
AA3	
AA2	num3 = 1
AA1	num2 = 2
AA0	num1 = 3

Pass By Value Example

The values stored in num1, num2 and num3 are going to be copied respectively

```
#include<stdio.h>

nction (int numl, int num2, int num3);

)

= 3;
int num2 = 2;
int num3 = 1;
printf (“num1 = %d\n", num1);
printf ("num2 = %d\n", num2);
printf ("num3 = %d\n", num3);

myFunction (num1, num2, num3);

printf ("num1 = %d\n", num1);
printf (“num2 = %d\n", num2) ;
printf ("num3 = %d\n", num3);
return 0;
}


void myFunction (int num1, int num2, int num3)
{
num1 = 5;
num2 = 8;

printf ("num1 = %d\n", num1);
printf ("num2 = %d\n" , num2);
printf ("num3 = %d\n", num3);
}
```

Here

Stack Space	
AA9	
AA8	
AA7	
AA6	
AA5	
AA4	
AA3	
AA2	num3 = 1
AA1	num2 = 2
AA0	num1 = 3

Pass By Value Example

Here

```
#include<stdio.h>

void myFunction (int numl, int num2, int num3);

int main()
{
int num1 = 3;
int num2 = 2;
int num3 = 1;
printf (“num1 = %d\n", num1);
printf ("num2 = %d\n", num2);
printf ("num3 = %d\n", num3);

myFunction (num1, num2, num3);

printf ("num1 = %d\n", num1);
printf (“num2 = %d\n", num2) ;
printf ("num3 = %d\n", num3);
return 0;
}


void myFunction (int num1, int num2, int num3)
{
num1 = 5;
num2 = 8;

printf ("num1 = %d\n", num1);
printf ("num2 = %d\n" , num2);
printf ("num3 = %d\n", num3);
}
```

Notice the parameters of the function are also num1, num2, and num3

Stack Space	
AA9	
AA8	
AA7	
AA6	
AA5	
AA4	
AA3	
AA2	num3 = 1
AA1	num2 = 2
AA0	num1 = 3

Pass By Value Example

The values of num1, num2, and num3 are going to be copied and stored respectively with the provided parameters

```
#include<stdio.h>

void myFunction (int numl, int num2, int num3);

int main()
{
int num1 = 3;
int num2 = 2;
int num3 = 1;
printf (“num1 = %d\n", num1);
printf ("num2 = %d\n", num2);
printf ("num3 = %d\n", num3);

myFunction (num1, num2, num3);

printf ("num1 = %d\n", num1);
printf (“num2 = %d\n", num2) ;
printf ("num3 = %d\n", num3);
return 0;
}


void myFunction (int num1, int num2, int num3)
{
num1 = 5;
num2 = 8;

printf ("num1 = %d\n", num1);
printf ("num2 = %d\n" , num2);
printf ("num3 = %d\n", num3);
}
```

Here

Stack Space	
AA9	
AA8	
AA7	
AA6	
AA5	
AA4	
AA3	
AA2	num3 = 1
AA1	num2 = 2
AA0	num1 = 3

Pass By Value Example

Here

```
#include<stdio.h>

void myFunction (int numl, int num2, int num3);

int main()
{
int num1 = 3;
int num2 = 2;
int num3 = 1;
printf (“num1 = %d\n", num1);
printf ("num2 = %d\n", num2);
printf ("num3 = %d\n", num3);

myFunction (num1, num2, num3);

printf ("num1 = %d\n", num1);
printf (“num2 = %d\n", num2) ;
printf ("num3 = %d\n", num3);
return 0;
}


void myFunction (int num1, int num2, int num3)
{
num1 = 5;
num2 = 8;

printf ("num1 = %d\n", num1);
printf ("num2 = %d\n" , num2);
printf ("num3 = %d\n", num3);
}
```

Stack Space	
AA9	
AA8	
AA7	
AA6	
AA5	
AA4	
AA3	num1 = 3
AA2	num3 = 1
AA1	num2 = 2
AA0	num1 = 3

Pass By Value Example

Here

```
#include<stdio.h>

void myFunction (int numl, int num2, int num3);

int main()
{
int num1 = 3;
int num2 = 2;
int num3 = 1;
printf (“num1 = %d\n", num1);
printf ("num2 = %d\n", num2);
printf ("num3 = %d\n", num3);

myFunction (num1, num2, num3);

printf ("num1 = %d\n", num1);
printf (“num2 = %d\n", num2) ;
printf ("num3 = %d\n", num3);
return 0;
}


void myFunction (int num1, int num2, int num3)
{
num1 = 5;
num2 = 8;

printf ("num1 = %d\n", num1);
printf ("num2 = %d\n" , num2);
printf ("num3 = %d\n", num3);
}
```

Stack Space	
AA9	
AA8	
AA7	
AA6	
AA5	num3 = 1
AA4	num2 = 2
AA3	num1 = 3
AA2	num3 = 1
AA1	num2 = 2
AA0	num1 = 3

Pass By Value Example

Hold

Here

```c
#include<stdio.h>

void myFunction (int numl, int num2, int num3);

int main()
{
int num1 = 3;
int num2 = 2;
int num3 = 1;
printf (“num1 = %d\n", num1);
printf ("num2 = %d\n", num2);
printf ("num3 = %d\n", num3);

myFunction (num1, num2, num3);

printf ("num1 = %d\n", num1);
printf (“num2 = %d\n", num2) ;
printf ("num3 = %d\n", num3);
return 0;
}


void myFunction (int num1, int num2, int num3)
{
num1 = 5;
num2 = 8;

printf ("num1 = %d\n", num1);
printf ("num2 = %d\n" , num2);
printf ("num3 = %d\n", num3);
}
```

Stack Space	
AA9	
AA8	
AA7	
AA6	
AA5	num3 = 1
AA4	num2 = 2
AA3	num1 = 3
AA2	num3 = 1
AA1	num2 = 2
AA0	num1 = 3

Pass By Value Example

```
#include<stdio.h>

void myFunction (int numl, int num2, int num3);

int main()
{
int num1 = 3;
int num2 = 2;
int num3 = 1;
printf (“num1 = %d\n", num1);
printf ("num2 = %d\n", num2);
printf ("num3 = %d\n", num3);

myFunction (num1, num2, num3);

printf ("num1 = %d\n", num1);
printf (“num2 = %d\n", num2) ;
printf ("num3 = %d\n", num3);
return 0;
}


void myFunction (int num1, int num2, int num3)
{
num1 = 5;
num2 = 8;

printf ("num1 = %d\n", num1);
printf ("num2 = %d\n" , num2);
printf ("num3 = %d\n", num3);
}
```

Hold

Here

The value in red text is now the variable being modified

Stack Space	
AA9	
AA8	
AA7	
AA6	
AA5	num3 = 1
AA4	num2 = 2
AA3	num1 = 5
AA2	num3 = 1
AA1	num2 = 2
AA0	num1 = 3

Pass By Value Example

```
#include<stdio.h>

void myFunction (int numl, int num2, int num3);

int main()
{
int num1 = 3;
int num2 = 2;
int num3 = 1;
printf (“num1 = %d\n", num1);
printf ("num2 = %d\n", num2);
printf ("num3 = %d\n", num3);

myFunction (num1, num2, num3);

printf ("num1 = %d\n", num1);
printf (“num2 = %d\n", num2) ;
printf ("num3 = %d\n", num3);
return 0;
}


void myFunction (int num1, int num2, int num3)
{
num1 = 5;
num2 = 8;

printf ("num1 = %d\n", num1);
printf ("num2 = %d\n" , num2);
printf ("num3 = %d\n", num3);
}
```

Hold

Here

The value in red text is now the variable being modified

Stack Space	
AA9	
AA8	
AA7	
AA6	
AA5	num3 = 1
AA4	num2 = 8
AA3	num1 = 5
AA2	num3 = 1
AA1	num2 = 2
AA0	num1 = 3

Pass By Value Example

```
#include<stdio.h>

void myFunction (int numl, int num2, int num3);

int main()
{
int num1 = 3;
int num2 = 2;
int num3 = 1;
printf (“num1 = %d\n", num1);
printf ("num2 = %d\n", num2);
printf ("num3 = %d\n", num3);

myFunction (num1, num2, num3);

printf ("num1 = %d\n", num1);
printf (“num2 = %d\n", num2) ;
printf ("num3 = %d\n", num3);
return 0;
}


void myFunction (int num1, int num2, int num3)
{
num1 = 5;
num2 = 8;

printf ("num1 = %d\n", num1);
printf ("num2 = %d\n" , num2);
printf ("num3 = %d\n", num3);
}
```

Hold

Here

Stack Space	
AA9	
AA8	
AA7	
AA6	
AA5	num3 = 1
AA4	num2 = 8
AA3	num1 = 5
AA2	num3 = 1
AA1	num2 = 2
AA0	num1 = 3

Pass By Value Example

```
#include<stdio.h>

void myFunction (int numl, int num2, int num3);

int main()
{
int num1 = 3;
int num2 = 2;
int num3 = 1;
printf (“num1 = %d\n", num1);
printf ("num2 = %d\n", num2);
printf ("num3 = %d\n", num3);

myFunction (num1, num2, num3);

printf ("num1 = %d\n", num1);
printf (“num2 = %d\n", num2) ;
printf ("num3 = %d\n", num3);
return 0;
}


void myFunction (int num1, int num2, int num3)
{
num1 = 5;
num2 = 8;

printf ("num1 = %d\n", num1);
printf ("num2 = %d\n" , num2);
printf ("num3 = %d\n", num3);
}
```

Hold

Here

Stack Space	
AA9	
AA8	
AA7	
AA6	
AA5	num3 = 1
AA4	num2 = 8
AA3	num1 = 5
AA2	num3 = 1
AA1	num2 = 2
AA0	num1 = 3

Pass By Value Example

```
#include<stdio.h>

void myFunction (int numl, int num2, int num3);

int main()
{
int num1 = 3;
int num2 = 2;
int num3 = 1;
printf (“num1 = %d\n", num1);
printf ("num2 = %d\n", num2);
printf ("num3 = %d\n", num3);

myFunction (num1, num2, num3);

printf ("num1 = %d\n", num1);
printf (“num2 = %d\n", num2) ;
printf ("num3 = %d\n", num3);
return 0;
}


void myFunction (int num1, int num2, int num3)
{
num1 = 5;
num2 = 8;

printf ("num1 = %d\n", num1);
printf ("num2 = %d\n" , num2);
printf ("num3 = %d\n", num3);
}
```

Hold

Here

Now we have reached the end of the user-defined function!

Stack Space	
AA9	
AA8	
AA7	
AA6	
AA5	num3 = 1
AA4	num2 = 8
AA3	num1 = 5
AA2	num3 = 1
AA1	num2 = 2
AA0	num1 = 3

Pass By Value Example

```
#include<stdio.h>

void myFunction (int numl, int num2, int num3);

int main()
{
int num1 = 3;
int num2 = 2;
int num3 = 1;
printf (“num1 = %d\n", num1);
printf ("num2 = %d\n", num2);
printf ("num3 = %d\n", num3);

myFunction (num1, num2, num3);

printf ("num1 = %d\n", num1);
printf (“num2 = %d\n", num2) ;
printf ("num3 = %d\n", num3);
return 0;
}


void myFunction (int num1, int num2, int num3)
{
num1 = 5;
num2 = 8;

printf ("num1 = %d\n", num1);
printf ("num2 = %d\n" , num2);
printf ("num3 = %d\n", num3);
}
```

Hold

Here

Now we have reached the end of the user-defined function!

Stack Space	
AA9	
AA8	
AA7	
AA6	
AA5	num3 = 1
AA4	num2 = 8
AA3	num1 = 5
AA2	num3 = 1
AA1	num2 = 2
AA0	num1 = 3

Pass By Value Example

```
#include<stdio.h>

void myFunction (int numl, int num2, int num3);

int main()
{
int num1 = 3;
int num2 = 2;
int num3 = 1;
printf (“num1 = %d\n", num1);
printf ("num2 = %d\n", num2);
printf ("num3 = %d\n", num3);

myFunction (num1, num2, num3);

printf ("num1 = %d\n", num1);
printf (“num2 = %d\n", num2) ;
printf ("num3 = %d\n", num3);
return 0;
}


void myFunction (int num1, int num2, int num3)
{
num1 = 5;
num2 = 8;

printf ("num1 = %d\n", num1);
printf ("num2 = %d\n" , num2);
printf ("num3 = %d\n", num3);
}
```

Here

After the function is done, it’s variables/ parameters are removed from the stack space.

Stack Space	
AA9	
AA8	
AA7	
AA6	
AA5	
AA4	
AA3	
AA2	num3 = 1
AA1	num2 = 2
AA0	num1 = 3

Pass By Value Example

```
#include<stdio.h>

void myFunction (int numl, int num2, int num3);

int main()
{
int num1 = 3;
int num2 = 2;
int num3 = 1;
printf (“num1 = %d\n", num1);
printf ("num2 = %d\n", num2);
printf ("num3 = %d\n", num3);

myFunction (num1, num2, num3);

printf ("num1 = %d\n", num1);
printf (“num2 = %d\n", num2) ;
printf ("num3 = %d\n", num3);
return 0;
}


void myFunction (int num1, int num2, int num3)
{
num1 = 5;
num2 = 8;

printf ("num1 = %d\n", num1);
printf ("num2 = %d\n" , num2);
printf ("num3 = %d\n", num3);
}
```

Here

After the function is done, it's variables/ parameters are removed from the stack space.

Stack Space	
AA9	
AA8	
AA7	
AA6	
AA5	
AA4	
AA3	
AA2	num3 = 1
AA1	num2 = 2
AA0	num1 = 3

Pass By Value Example

```
#include<stdio.h>

void myFunction (int numl, int num2, int num3);

int main()
{
int num1 = 3;
int num2 = 2;
int num3 = 1;
printf (“num1 = %d\n", num1);
printf ("num2 = %d\n", num2);
printf ("num3 = %d\n", num3);

myFunction (num1, num2, num3);

printf ("num1 = %d\n", num1);
printf (“num2 = %d\n", num2) ;
printf ("num3 = %d\n", num3);
return 0;
}


void myFunction (int num1, int num2, int num3)
{
num1 = 5;
num2 = 8;

printf ("num1 = %d\n", num1);
printf ("num2 = %d\n" , num2);
printf ("num3 = %d\n", num3);
}
```

Here

After the function is done, it’s variables/ parameters are removed from the stack space.

Stack Space	
AA9	
AA8	
AA7	
AA6	
AA5	
AA4	
AA3	
AA2	num3 = 1
AA1	num2 = 2
AA0	num1 = 3

Pass By Value Example

```
#include<stdio.h>

void myFunction (int numl, int num2, int num3);

int main()
{
int num1 = 3;
int num2 = 2;
int num3 = 1;
printf (“num1 = %d\n", num1);
printf ("num2 = %d\n", num2);
printf ("num3 = %d\n", num3);

myFunction (num1, num2, num3);

printf ("num1 = %d\n", num1);
printf (“num2 = %d\n", num2) ;
printf ("num3 = %d\n", num3);
return 0;
}


void myFunction (int num1, int num2, int num3)
{
num1 = 5;
num2 = 8;

printf ("num1 = %d\n", num1);
printf ("num2 = %d\n" , num2);
printf ("num3 = %d\n", num3);
}
```

Here

After the function is done, it’s variables/ parameters are removed from the stack space.

Stack Space	
AA9	
AA8	
AA7	
AA6	
AA5	
AA4	
AA3	
AA2	num3 = 1
AA1	num2 = 2
AA0	num1 = 3

Acknowledgements

Slides adapted from Dr. Andrew Steinberg's COP 3223H course