

```
// structure
typedef struct dynarrStruct
{
    int* arrptr;
    int allocated;
    int inUse;
} dynarr;

From <https://www.cs.uic.edu/pub/CS211/NotesS18/CS\_211\_-\_3918.pdf>
void init ( dynarr* arr)
{ // initially allocate space for 10 values
    arr->allocated = 10;
    arr->arrptr = ( int* ) malloc ( sizeof(int) *
                                arr->allocated ) ;
    arr->inUse = 0;
}
```

```
void addValue ( dynarr* arr, int val)
{
    if ( arr->allocated <= arr->inUse )
    {
        int* tmp = ( int* ) malloc ( sizeof(int) *
                                    arr->allocated * 2 ) ;

        int a;
        for ( a = 0 ; a < arr->inUse ; a++)
            tmp[a] = arr->arrptr[a];

        free ( arr->arrptr );

        arr->arrptr = tmp;
        arr->allocated = arr->allocated * 2;
```

```
// class
class Dynarr
{
    private:
        int* arrptr;
        int allocated;
        int inUse;

    public:
        Dynarr ( )
        { // initially allocate space for 10 values
            allocated = 10;
            //this->arrptr = ( int* ) malloc ( sizeof(int)
            //      * this->allocated ) ;
            this->arrptr = new int [ this->allocated ] ;
            inUse = 0;
        }
```

```
void init ( ) // replaced by constructor
{ // initially allocate space for 10 values
    allocated = 10;
    //this->arrptr = ( int* ) malloc ( sizeof(int)
    //      * this->allocated ) ;
    this->arrptr = new int [ this->allocated ] ;
    inUse = 0;
}
```

```
void addValue ( int val)
{
    if ( allocated <= inUse )
    {
        //int* tmp = ( int* ) malloc ( sizeof(int) *
        //      allocated * 2 ) ;
        int* tmp = new int [allocated * 2];
        int a;
        for ( a = 0 ; a < inUse ; a++)
            tmp[a] = arrptr[a];

        //free ( arrptr );
        delete [] arrptr;

        arrptr = tmp;
        allocated = allocated * 2;
```

```
}  
arr->arrptr[arr->inUse] = val;  
arr->inUse = arr->inUse + 1;  
}
```

From <[https://www.cs.uic.edu/pub/CS211/NotesS18/CS_211 - 3918.pdf](https://www.cs.uic.edu/pub/CS211/NotesS18/CS_211_-_3918.pdf)>

```
main()  
{  
  dynarr arr1;  
  init ( & arr1);  
  addValue ( & arr1, val );
```

From <[https://www.cs.uic.edu/pub/CS211/NotesS18/CS_211 - 3918.pdf](https://www.cs.uic.edu/pub/CS211/NotesS18/CS_211_-_3918.pdf)>

```
}  
arrptr[inUse] = val;  
inUse = inUse + 1;  
}  
  
};
```

From <[https://www.cs.uic.edu/pub/CS211/NotesS18/CS_211 - 3918.pdf](https://www.cs.uic.edu/pub/CS211/NotesS18/CS_211_-_3918.pdf)>

```
main()  
{  
  Dynarr arr1; // automatically call the default constructor  
  //arr1.init ( ); // called by constructor - not needed here  
  arr1.addValue ( val );
```