Lecture 2 b

2D Dynamic Array

Static memory 1d array

```
void printNumbers(int num[],int len){

    for(int i = 0; i < len; ++i){

         cout<<num[i]<<endl;
     }
}

int main(){

    const int N = 4;

    int numArr[N] = {1,2,3,4};

    printNumbers(numArr, N);

    return 0;
}
```

Dynamic memory 1d array

```
void printNumbers(int *num,int len){

    for(int i = 0; i < len; ++i){

           cout<<num[i]<<endl;
     }
}

int main(){

    int N = 4;

    int *numArr = new int[N]{1,2,3,4};

    printNumbers(numArr, N);

    delete[] numArr;

    return 0;
}
```

Static memory 2d array

```
void printGrades(int grades[][4],int nSubjs,int nYears){

    for(int i = 0; i < nYears; ++i){

        for(int j = 0; j < nSubjs; ++j){

            cout << grades[i][j] << " ";

        }
        cout<<endl;
    }
}
```

Static memory 2d array

```
int main(){

  const int nYears = 2;
  const int nSubjs = 4;

  int grades[nYears][nSubjs] = {

                          {100,88,99,100},
                          {99,92,95,94}
                       };

  printGrades(grades, nSubjs, nYears);
  return 0;
}
```

Dynamic memory 2d array

```
void printGrades(int *grades[2],int nYears,int nSubjs){
    for(int i = 0; i < nYears; ++i){
       for(int j = 0; j < nSubjs; ++j){
          cout << grades[i][j] << " ";
        }
        cout << endl;
   }
}
```

Dynamic memory 2d array

```
int main(){

    const int nYears = 2;int nSubjs = 4;

    //Pointer allocation in stack

    int *grades[nYears];

    //Space allocation

    for(int i = 0; i < nYears; ++i){

        grades[i] = new int[nSubjs];
    }
```

Dynamic memory 2d array

```
//Value initialization
for(int i = 0; i < nYears; ++i){
  for(int j = 0; j < nSubjs; ++j){
      grades[i][j] = i*i;
  }
}
```

Dynamic memory 2d array

```
   printGrades(grades,nYears,nSubjs);

   //Free memory

   for(int i = 0; i < nYears; ++i){

     delete[] grades[i];

   }

   return 0;

}
```

Dynamic memory 2d array

```
void printGrades(int** grades, int nYears, int nSubjs) {

       for (int i = 0; i < nYears; ++i) {

              for (int j = 0; j < nSubjs; ++j) {

                     cout << grades[i][j] << " ";

              }
              cout << endl;
       }
}
```

Dynamic memory 2d array

```
int main(){
        int nYears = 2; int nSubjs = 4;
         //Pointer allocation in heap
        int** grades = new int* [nYears];
        //Space allocation
        for (int i = 0; i < nYears; ++i) {
                grades[i] = new int[nSubjs];
        }
```

Dynamic memory 2d array

```
//Value initialization

for (int i = 0; i < nYears; ++i) {

       for (int j = 0; j < nSubjs; ++j) {

              grades[i][j] = i;

       }

}
```

Dynamic memory 2d array

```
printGrades(grades, nYears, nSubjs);
//Free memory
for (int i = 0; i < nYears; ++i){
       delete[] grades[i];
}
return 0;
}
```

Handling Errors

```
#include <iostream>

using namespace std;

int main() {

          int nYears = 100000000000;

          int* arr = new (nothrow) int[nYears];

          if (arr == NULL) {

                    cout << "Error" << endl;

          }

          return 0;

}
```