IntelliJ Debugger

Memory issues

       Example code: "Factorial" (CLion)

       Demonstrate:

              Stack

              "DataArea" with and without allocating memory

                     Declared without allocating data area

                     Contents of DataArea

                     Delete: incrementing pointer

                     Why "new" instead of "malloc"?

Templates may seem intimidating at first glance. However, all they represent is a shorthand that writes code for you.
Since they do not create code themselves, they generally belong in your header files.

Template "GetMax"
```cpp
//==========================================
//==========================================
template <class userType>
userType GetMax (userType a, userType b) {
    return (a>b?a:b);
}
```

Linked Lists

Code Examples: RasterClass (Online GDB)

Singly Linked List:
```
void InsertAfter(charRaster* newRast) {
    newRast->next = next;
    next = newRast;
}
```
https://visualgo.net/en/list

DOUBLY linked list:
```
void InsertAfter(charRaster* newRast) {
    newRast->next = next;
    newRast->prev = this;
    if (next!=NULL)
        next->prev = newRast;
    next = newRast;
}
```

Rule of Zero

Don't write any constructors or destructors if you don't need them.

Rule of Five

The rule of five is a modern expansion of the rule of three. Firstly, the *rule of three* specifies that if a class implements any of the following functions, it should implement all of them:

- copy constructor (deep copy)
- copy assignment operator
- Destructor

Rule of Five adds:
- Move Constructor
- Move Assignment Operator

The "move" differs from the "copy" - Rather than copy the resource, they take the resource from the original object and set its internal pointer to NULL, effectively stealing the resource.

Naive Bayes
https://courses.grainger.illinois.edu/cs126/sp2020/assignments/naivebayes-example.pdf