Jaap van der Leest
Timon van der Berg
October 21, 2017

C++ Course
Assignment 6

Exercise 42

Problem statement. *What are the variants of new/delete? For each of the variants provide a (short!) example in which the used new/delete is appropriate and provide a short explanation why it is appropriately used.*
Solution.

new/delete

`new` is used to allocate memory for a primitive type or object. When allocating for an object it will call the object constructor. Example of `new` use:

```
int *ptr = new int;
```

This allocation is appropriate since we want to allocate memory for a single int.

`delete` is used to deallocate memory that was allocated using `new`. If called on an object (not a primitive type) it will also call that objects' destructor. Example of `delete` use:

```
std::string *ptr = new std::string;
delete ptr;
```

This is appropriate because `delete` is used on memory allocated using `new`.

new[], delete[]

`new[]` is used to allocate memory for arrays. Like `new` it is type-safe: the type of the element has to be declared. Like `new`, it calls constructors. An example of using `new`:

```
int *aoi = new int[20];
```

`delete[]` is used to delete memory allocated using `new[]`. Unlike `new`, new[] saves the size of the array it allocates. `delete[]` uses this to delete the array. Destructors are called[1]. An example of `delete[]` usage:

```
string *strp = new string[550];
delete[] strp;
```

This is appropriate because `delete[]` is used on an array allocated using `new[]`.

operator new, operator delete

`operator new` is used to allocate raw bytes of memory. To actually use this memory, a static cast is required. An example of using `operator new`:

```
size_t *sp = static_cast<size_t *>(operator new(5 * sizeof(size_t)));
```

This is appropriate because `operator new` is used to allocate raw memory. Here we first calculate the number of bytes needed for 5 `size_t` variables. We then allocate the memory. `operator new` does not care for types.

`operator delete` is used to deallocate memory that was allocated using `operator new`. Like `operator new`, `operator delete` does not care for types. Because `operator new` saves the number of bytes allocated, `operator delete` knows how much memory to deallocate. `operator delete` does not call any destructors. An example of using `operator delete`:

[1] If the array contains a primitive type no destructors are called. Therefore an array of pointers require manual destruction of whatever is pointed to.

```
string *sp = static_cast<string *>(operator new(5 * sizeof(string)));
operator delete(sp)
```

This is appropriate because we are using operator delete to deallocate memory that was allocated using operator new.

placement new

placement `new` is found in `<memory>` and overloads `new`. Placement `new` is used to place objects in previously allocated memory. An example of using placement `new`:

```
string *sp = static_cast<string *>( operator new(15 * sizeof(string)));
new sp string("Donald Knuth");
```

This is appropriate because we are using operator new on memory of the correct type that was previously allocated. We have placed a single string in this memory, leaving room for 14 more.