

Common memory related errors in C++

Most of these errors can be detected using a dynamic program analysis tool, such as **valgrind**.

Uninitialized Pointer

Using a pointer before being initialized with a valid address, causing it to point to an unknown, arbitrary memory location.

```
int *p;
```

```
*p = 7; //  p points to unknown location
```

Invalid Address Access

Trying to read or write to a memory location that is unmapped, restricted, or outside its authorized address space.

```
int *p = nullptr;
```

```
*p = 7; //  null pointer dereference
```

Memory Leak

Forgetting to free heap-allocated memory, leaving it occupied and inaccessible for the remainder of the program's execution.

```
int *p = new int {10};
```

```
//  memory leak, forgot to delete
```

Buffer Overflow

Writing data past the allocated boundaries of a memory buffer, which can corrupt adjacent data.

```
int arr[3];
```

```
arr[3] = 42;
```

```
//  out of bounds (valid indexes: 0,1,2)
```

Use-After-Free

Accessing heap-allocated memory after it has been deleted or deallocated.

```
int *p = new int {10};
```

```
delete p;
```

```
*p = 5; //  dangling pointer
```

Double Free

Attempting to delete the same heap-allocated memory location more than once, which corrupts the memory manager's internal state.

```
int *p = new int {10};
```

```
delete p;
```

```
delete p; //  double delete
```