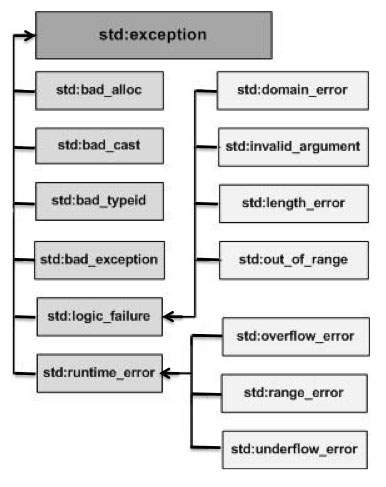
C++ Standard Exceptions

C++ provides a list of standard exceptions defined in **<exception>** which we can use in our programs. These are arranged in a parent-child class hierarchy shown below −



Here is the small description of each exception mentioned in the above hierarchy −

|  |  |
| --- | --- |
| **Sr.No** | **Exception & Description** |
| 1 | **std::exception**  An exception and parent class of all the standard C++ exceptions. |
| 2 | **std::bad\_alloc**  This can be thrown by **new**. |
| 3 | **std::bad\_cast**  This can be thrown by **dynamic\_cast**. |
| 4 | **std::bad\_exception**  This is useful device to handle unexpected exceptions in a C++ program. |
| 5 | **std::bad\_typeid**  This can be thrown by **typeid**. |
| 6 | **std::logic\_error**  An exception that theoretically can be detected by reading the code. |
| 7 | **std::domain\_error**  This is an exception thrown when a mathematically invalid domain is used. |
| 8 | **std::invalid\_argument**  This is thrown due to invalid arguments. |
| 9 | **std::length\_error**  This is thrown when a too big std::string is created. |
| 10 | **std::out\_of\_range**  This can be thrown by the 'at' method, for example a std::vector and std::bitset<>::operator[](). |
| 11 | **std::runtime\_error**  An exception that theoretically cannot be detected by reading the code. |
| 12 | **std::overflow\_error**  This is thrown if a mathematical overflow occurs. |
| 13 | **std::range\_error**  This is occurred when you try to store a value which is out of range. |
| 14 | **std::underflow\_error**  This is thrown if a mathematical underflow occurs. |