CSE 333  Section 7 - Casting

Welcome back to section! We’re glad that you’re here :)  

Casting in C++

While in C++, we want to use casts that are more explicit in their behaviour. This gives us a better understanding of what happens when we read our code, because C-style casts can do many (sometimes unwanted) things. There are four types of casts we will use in C++:

\[
\text{static\_cast<to\_type>(expression);} \\
\quad \star\text{ Converts between pointers of related types.} \\
\quad \quad \circ\text{ Compiler error if not related.} \\
\quad \star\text{ Performs not pointer conversion (e.g. float to int conversion).}
\]

\[
\text{dynamic\_cast<to\_type>(expression);} \\
\quad \star\text{ Converts between pointers of related types.} \\
\quad \quad \circ\text{ Compiler error if not related.} \\
\quad \circ\text{ Also checks at runtime to make sure it is a 'safe' conversion (returns nullptr if not).}
\]

\[
\text{const\_cast<to\_type>(expression);} \\
\quad \star\text{ Used to add or remove const-ness.}
\]

\[
\text{reinterpret\_cast<to\_type>(expression);} \\
\quad \star\text{ Casts between incompatible types without changing the data.} \\
\quad \circ\text{ The types you are casting to and from must be the same size.} \\
\quad \circ\text{ Will not let you convert between integer and floating point types.}
\]

Exercise 1

For each of the following snippets of code, fill in the blank with the most appropriate C++ style cast. Assume that we have the following classes defined:

\[
\begin{align*}
\text{class Base } & : \text{ public Base } \\
\{ & \\
\text{ public:} & \\
\text{ int } x; & \\
\}; & \\
\text{class Derived : public Base } & : \text{ public Base } \\
\{ & \\
\text{ public:} & \\
\text{ int } y; & \\
\};
\end{align*}
\]

\[
\begin{align*}
t64_t x & = 0x7fffffffffe870; \\
\text{char* str} & = \text{reinterpret\_cast<char *}(x);} \\
\end{align*}
\]

\[
\begin{align*}
\text{void foo(Base *b)} & \{ \\
\text{ Derived *d} & = \text{dynamic\_cast<Derived *>(b);} \\
\text{ // additional code omitted} & \}
\end{align*}
\]

\[
\begin{align*}
\text{Derived *d} & = \text{new Derived;} \\
\text{Base *b} & = \text{static\_cast<Base *>(d);} \\
\end{align*}
\]

\[
\begin{align*}
\text{double x} & = 64.382; \\
\text{int64_t y} & = \text{static\_cast<int64_t>(x);} \\
\end{align*}
\]