Lab 8: Object Layout and Vtables

Learning objectives:

- Understand object layout with inheritance
- Understand vtable layout in object inheritance hierarchies

This lab does not have any distribution code.

1. *Vtables.* Consider the following C++ code:

```
struct A {
  void foo() {
    cout << "A::foo()" << endl;
  }
  virtual void bar() {
    cout << "A::bar()" << endl;
  }
};

struct B : A {
  virtual void foo() {
    cout << "B::foo()" << endl;
  }
  void bar() {
    cout << "B::bar()" << endl;
  }
};

int main() {
  A *aptr = new B;
  aptr->foo();
  aptr->bar();
}
```

This code prints the following when run:

```
A::foo()
B::bar()
```

a) Draw the vtables for `A` and `B`. Clearly indicate the name of each member as well as which type contains its definition (e.g. `T::member`).

b) Briefly explain how the compiler translates the method calls in `main()`.

2. *Vtable and object layout.* Consider the following C++ code:

```
#include <iostream>

using std::cout;
using std::endl;

struct A {
  int x;
  virtual void spam() { cout << "A::spam()" << endl; }
```

```
  virtual void eggs() { cout << "A::eggs()" << endl; }
};

struct B : A {
  int x;
  virtual void eggs() { cout << "B::eggs()" << endl; }
};

struct C : B {
  int z;
  virtual void spam() { cout << "C::spam()" << endl; }
};
```

a) Draw a picture illustrating the contents of an **object** of type `C`. Clearly indicate what each entry in the object is, and if it is a member, which type contains its definition (e.g. `T::member`).

b) Draw the layout of the **vtable** for type `B`. Clearly indicate the name of each member as well as which type contains its definition.

c) Draw the layout of the **vtable** for type `C`. Clearly indicate the name of each member as well as which type contains its definition.