   a. Define the following terms:
      i. Static Binding
      ii. Dynamic Binding
      iii. Overriding (a function)
      iv. Overloading (a function)
      v. Redefining (a function)
   b. For this part, reference the following code below and fill in the 4 blanks with what would print on each of these lines and the type of binding that is performed (iostream/std namespace are implied):

   ```cpp
   class Vehicle {
   public:
      void f_1(){ cout << "Vehicle f_1" << endl; }
      virtual void f_2{ cout << "Vehicle f_2" << endl; }};
   class Truck : public Vehicle {
   public:
      void f_1(){ cout << "Truck f_1" << endl; }
      virtual void f_2(){ cout << "Truck f_2" << endl; }};
   int main(){
      Truck t;
      Vehicle *p_vs = &t;
      Vehicle *p_vd = new Truck;
      p_vs->f_1(); // i.)
      p_vs->f_2(); // ii.)
      p_vd->f_1(); // iii.)
      p_vd->f_2(); // iv.)
      return 0; }
   ```

2. What is an abstract class? Provide an example that has not been used in class and provide an adequate explanation why you would design it as an abstract class with at least two example subclasses that would derive from it. What is a pure virtual function? Where / how (syntax) are they used?

3. Why might it be important to create a virtual destructor for a base class? Do you need a destructor for an abstract class?