ET-580 - Polymorphism & Virtual Functions - Practice

1. Implement the following:

   a. Class named *Person*
      1. data member: *name*
      2. output function: output the *name*

   b. Derived class named *Student* that inherits from *Person*
      1. data member: *id* (integer)
      2. output function: redefine to print *name* and *id*

   c. Derived class named *Instructor* that inherits from *Person*
      1. data member: *department*
      2. output function: redefine to print *name*, and *department*

   d. In main implement a *Person* pointer named *p*

   e. Assign *p* to a new dynamic variable of type *Person*, call output

   f. Assign *p* to a new dynamic variable of type *Student*, call output

   g. Assign *p* to a new dynamic variable of type *Instructor*, call output

   Note that the *Person* output function is called for each object.

Example Output

```
Joseph
Dion
Mr. Evans
```

2. Clone the previous program, and implement:

   a. Class *Person*
      1. make output a virtual function

   b. Class *Student*
      1. add override modifier to output

   c. Class *Instructor*
      1. add override modifier to output

   d. Run the program.

   Note that the appropriate output function is called for each object.

Example Output

```
Joseph
Dion, 1534442
Mr. Evans, Comp. Sci
```

3. Clone the previous program, and implement:

   a. Change *Person* into an abstract class.
   b. Remove the code which instantiates a Person object in main.
   c. Run the program.

Note that we can only instantiate derived objects of abstract classes.

Example Output

```
Dion, 1534442
Mr. Evans, Comp. Sci
```

4. Clone the previous program, and implement:

   a. A non-member *print* function.
      This function accepts an object of type *Person* or *Person* subtype.
      This function calls the appropriate *output* function for the object.
   b. In main replace all *output* function calls with non-member *print* function calls.
   c. Test the program.

Note that we can maintain polymorphism using pass by reference or pointer.

Example Output

```
Dion, 1534442
Mr. Evans, Comp. Sci
```