Data Structures
CS 246 - 040
Department of Physics and Computer Science
Medgar Evers College
Exam 1 Redo

Instructions:

- Create the header file '`Game.h`' that contain a header guard and includes the libraries *iostream*, *string*, *sstream*, and *iomanip*.
- Your submissions must be submitted to the GitHub repository in the Exam01 directory.
- Cheating of any kind is prohibited and will not be tolerated.
- Violating or failing to follow any of the rules above will result in an automatic zero (0) for the lab.

TO ACKNOWLEDGE THAT YOU HAVE READ AND UNDERSTOOD THE INSTRUCTIONS ABOVE, PRINT YOUR NAME AND THE DATE ON BOTH THIS SHEET AND THE BLUE BOOK

Name: _____ **Date:** _____

Grading

Task	Maximum Points	Points Earned
00	0.10	
01	0.10	
02	0.10	
03	0.10	
04	0.10	
05	0.50	
06	0.50	
07	1.50	
08	2.00	
Total	5	

Within a namespace named *dse* (task 00), do the following

1. declare the class *Game*.
2. declare the class *Player*.
3. declare the class *Card*.
4. declare the class *Creature*.
5. define *Game* that contains
   - private *Player* pointer array field named *players* with size 2.
   - private *Card* pointer array field named *field* with size 16.
   - private integer array field named *occupants* with size 2.
   - private deleted copy constructor.
   - private deleted assignment operator.
   - public default constructor that assigns null to each element of *players* and *field*, and 0 to each element of *occupants*.
6. define *Player* that contains
   - private int field named *life*.
   - private int field named *mana*.
   - private *Card* pointer array field named *deck* with size 50.
   - private deleted copy constructor.
   - private deleted assignment operator.
   - public default constructor that assigns null to each element of *deck*, 5000 to *life*, and 100 to *mana*.
7. define *Card* that contains
   - private string field named *name*.
   - private integer field named *cost*.
   - private integer field named *type*.
   - private integer field named *turns*.
   - private Boolean field named *playable*.
   - private *Game* pointer field named *areana*.
   - public default constructor that initializes *name*, *cost*, *type*, *turns*, *playable*, and *arena* to `"nobody"`, 0, 1, 0, `false`, and null, respectively.
   - public overloaded constructor takes a string parameter and two integer parameters, respectively, and initializes *name*, *cost*, *type*, *turns*, *playable*, and *arena* to the first parameter, second parameter, third parameter, 0, `false`, and null, respectively.
   - public copy constructor that performs a shallow copy.
   - public assignment operator that performs a shallow copy.
   - public pure virtual void method named `ability()` that takes no parameters.
   - public virtual string constant method named `category()` that takes no parameters and returns the string `"unknown"`.
   - public constant getter method named `title()` for *name*.
   - public constant getter method named `value()` for *cost*.
   - public constant getter method named `group()` for *type*.
   - public constant getter method named `runs()` for *turns*.
   - public constant getter method named `active()` for *playable*.
8. define *Creature* that publicly inherits *Card* and contains
   - private integer field named *attack*.
   - private integer field named *defense*.
   - private integer field named *life*.
   - private integer field named *mana*.
   - public default constructor that initializes *name*, *cost*, *type*, *turns*, *playable*, *arena*, *attack*, *defense*, *life*, and *mana* to `"nobody"`, 0, 1, 0, `false`, null, 100, 100, 500, and 20, respectively.

- public overloaded constructor takes a string parameter and four integer parameters, respectively, and initializes *name*, *cost*, *type*, *attack*, *defense*, *life*, *mana*, *turns*, *playable*, and *arena* to the first parameter, second integer parameter, third parameter, fourth parameter, fifth parameter, 500, 20, 0, false, and null, respectively.
- public copy constructor that performs a shallow copy.
- public assignment operator that performs a shallow copy.
- public overridden `category()` method that that returns `"Dragon"`, `"Elf"`, `"Knight"`, `"Mage"`, `"Beast"`, `"Elemental"` and `"Fairy"` for *type* values 1 through 7, respectively.
- public string constant method `toString()` that takes no parameters and returns a string in the format

  `"`*n*`[`*t*`]\n`*l*` (`*m*`)\nATK/`*a*` DEF/`*d*` CST/`*c*`"`

  where *n*, *t*, *l*, *m*, *a*, *d*, and *c* are the values of *name*, *type*, *life*, *mana*, *attack*, *defense*, and *cost*, respectively.