//---
//
//------------------------ Adventure Fighter----------------------
//
//---
//
// Program Name: Enemy Class Header
//
// Author : Mosfiqur Rahman
// Date : June, 2016
// Last Modified: 2nd June, 2016

- **System Manual:**
  - The system uses the Enemy base class (see Enemy.h and Enemy.cpp files)
  - The current system uses the following methods and attribute from the Enemy class:

    - Default [***Constructor***]:
      - **Enemy()**: it initializes the condition of the enemy
      - **virtual ~Enemy()**: destructor

    - Other methods: gets & sets the values of specific attributes
      - **virtual string getName() const = 0** :
        - ✔ gets the name of the enemy

      - **virtual string getDescription() const = 0**:
        - ✔ gets the description of th enemy

      - **virtual void attack(Enemy* hero) = 0**:
        - ✔ attacks other enemy

      - **virtual void defenseAttackShield(Enemy* hero) = 0**:
        - ✔ choices of the postions i.e. defense, attack or shield

      - **virtual string attackshield() const = 0**:
        - ✔ names of the postions i.e. shield & attack

      - **virtual void specialAttack(Enemy* hero) = 0**:
        - ✔ choices of the postions i.e. special attacks

      - **virtual string specialName() const = 0**:
        - ✔ names of the postions i.e. defense & attack

      - **virtual int getHealth() const = 0**:
        - ✔ gets the health of the enemy

- **virtual void dodamage(int damage) = 0**:
  - ✔ does damage

- **virtual void resetHealth() = 0**:
  - ✔ resets othe health

- **virtual void potionhealth() = 0**:
  - ✔ gains helath

// Program Name: Hero Class Header
//
// Author : Mosfiqur Rahman
// Date : June, 2016
// Last Modified: 2nd June, 2016

- **System Manual:**
  - The system uses the Enemy & Hero class (see Enemy.h, Hero.h, Enemy.cpp and Hero.cpp files)
  - The current system uses the following methods and attribute from the card class:

    - Private [***Attribute***]:
      - **string heroname**
        - ✔ suitable data structure for flawless programming operation
      - **string selectattack**
        - ✔ suitable data structure for flawless programming operation
      - **int potions**
        - ✔ suitable data structure for flawless programming operation
      - **int fireballs**
        - ✔ suitable data structure for flawless programming operation
      - **string input**
        - ✔ suitable data structure for flawless programming operation
      - **int heroHealth**
        - ✔ suitable data structure for flawless programming operation
      - **bool defense_mode**
        - ✔ suitable data structure for flawless programming operation
      - **string sheildinput**
        - ✔ suitable data structure for flawless programming operation

    - Default [***Constructor***]:
      - **Hero(string n)**: it initializes a Hero

    - Other methods: gets & sets the values of specific attributes
      - **int gettotalfireballs()** :

- ✔ gets the value of fireballs

- **int setattackfireballs(int)**:
  - ✔ initializes the amount of fireballs to attack

- **int setattackpotion(int)**:
  - ✔ sets the potion attack

- **void defenseAttackShield(Enemy * heroone)**:
  - ✔ choices of the postions i.e. defense, attack or shield

- **string attackshield()const**:
  - ✔ choices of the postions i.e. shield

- **string getName() const**:
  - ✔ gets the name of the hero

- **string getDescription() const**:
  - ✔ gets the description of the hero

- **void attack(Enemy * heroone)**:
  - ✔ attacks the enemy

- **string attackName() const**:
  - ✔ name of the attack

- **void specialAttack(Enemy * heroone)**:
  - ✔ special attack towards enemy

- **string specialName() const**:
  - ✔ name of the special attack

- **int getHealth() const**:
  - ✔ gets the health

- **void dodamage(int damage)**:
  - ✔ damages the enemy health

- **void resetHealth()**:
  - ✔ resets the health
- **void potionsattack(Enemy * heroone)**:
  - ✔ potion attack

- **void potionhealth()**:
  - ✔ potion health