```
#include <iostream>
#include <string>
#include <vector>

#include "dice.hpp"
#include "helpers.hpp"

using namespace std;

#pragma once

class Weapon {
  protected:
    string name;    // name of weapon
    string damage;  // damage per roll or "use"
    Dice*  die;

public:
    /**
     * @brief Construct a new Base Weapon object
     *  Also invokes the Dice class constructor with the "die" type
     */
    Weapon() {
        name = "Fists&Feet";
        damage = "1.d.4";
        die = new Dice(damage);
    }
    /**
     * @brief Overloaded Constructor for a new Base Weapon object
     *  Also invokes the Dice class constructor with the "die" type
     * @param choices-vector of string "die" choices like
{"2.d.8","2.d.10","3.d.4"}
     */
    Weapon(vector< string > choices) {
        name = "Fists&Feet";
        damage = randomChoice(choices);
        die = new Dice(damage);
    }

    virtual int use() {
        return die->roll();
    }

    // friend ostream& operator<<(ostream& os, const Weapon& w) {
    //     return os << "[" << w.name << " , " << w.damage << "]";
    // }
};
/*
*@brief: weapon "sword"
*@method: Sword() creates weapon type sword
*/
struct Sword : public Weapon{
```

```
  Sword(){
    name = "Sword";
    vector<string> roll = {"1.d.12", "2.d.6", "3.d.4"};
    damage = randomChoice(roll);
    die = new Dice(damage);
  }
};
/*
*@brief: weapon "Bow"
*@method: Bow() creates weapon type Bow
*/
struct Bow : public Weapon{
  Bow(){
    name = "Bow";
    vector<string> roll = {"1.d.8", "2.d.4", "1.d.10"};
    damage = randomChoice(roll);
    die = new Dice(damage);
  }
};
/*
*@brief: weapon "Spell"
*@method: Spell() creates weapon type Spell
*/
struct Spell : public Weapon{
  Spell(){
    name = "Magic Spell";
    vector<string> roll = {"1.d.20", "2.d.10", "3.d.6", "5.d.4"};
    damage = randomChoice(roll);
    die = new Dice(damage);
  }
};
/*
*@brief: weapon "M_Weapon"
*@method: M_Weapon() creates weapon type magic sword
*/
struct M_Weapon : public Weapon{
  M_Weapon(){
    name = "Magic Weapon";
    vector<string> roll1 = {"1.d.12", "2.d.6", "3.d.4"};
    vector<string> roll2 = {"1.d.4", "1.d.6"};
    damage = randomChoice(roll1) + randomChoice(roll2);
    die = new Dice(damage);
  }
};
/*
*@brief: weapon "F_Weapon" fire weapon
*@method: F_Weapon() creates weapon type fire weapon
*/
struct F_Weapon : public Weapon{
  F_Weapon(){
    name = "Magic Spell and Fire";
    vector<string> roll1 = {"1.d.20", "2.d.10", "3.d.6", "5.d.4"};
    vector<string> roll2 = {"1.d.6", "1.d.8"};
    damage = randomChoice(roll1) + randomChoice(roll2);
```

```
      die = new Dice(damage);
    }
  };
```