Explain the following:

1) the * operator (for declaring a “pointer”, and for “de-referencing”)
2) the & operator
3) the difference between a pointer and a reference in C++

1

`*` declares a pointer to something. Below `Dog *d4` is used to create a pointer to a dog object. Because `*d4` is a pointer to an object, it must be dereferneced to use to access the attributes of the dog class and use its functions.

```
// a pointer to a dog obj
Dog *d4 = new Dog("Bob");

// dereferencing
std::cout << (*d4).getName() << std::endl;

// syntactic sugar for the above dereferencing
std::cout << d4->getName() << std::endl;
```

2

`&` declares a reference to something. In `main.cpp` the line `Dog *dref = &d5;` will save the reference to the dog object d5 into the pointer dref. In order to dereference like in the above example, the following can be done.

```
Dog *dref = &d5;
std::cout << &d5 << std::endl;
std::cout << (*(&d5)).getName() << std::endl;
std::cout << (*dref).getName() << std::endl;
std::cout << dref << std::endl;
```

3 - pointer VS reference

`https://stackoverflow.com/questions/57483/what-are-the-differences-between-a-pointer-variabl`

- a pointer can be reassigned
- a reference cannot be reassigned
- a pointer has its own memory address
- a reference shares a memory address with the original variable
- a pointer can refer to another pointer (many layers of indirection)

- a reference can only have one level of indirection
- a pointer can be assigned null
- a reference cannot be assigned null
- a pointer uses `->` to access members
- a reference uses `.`