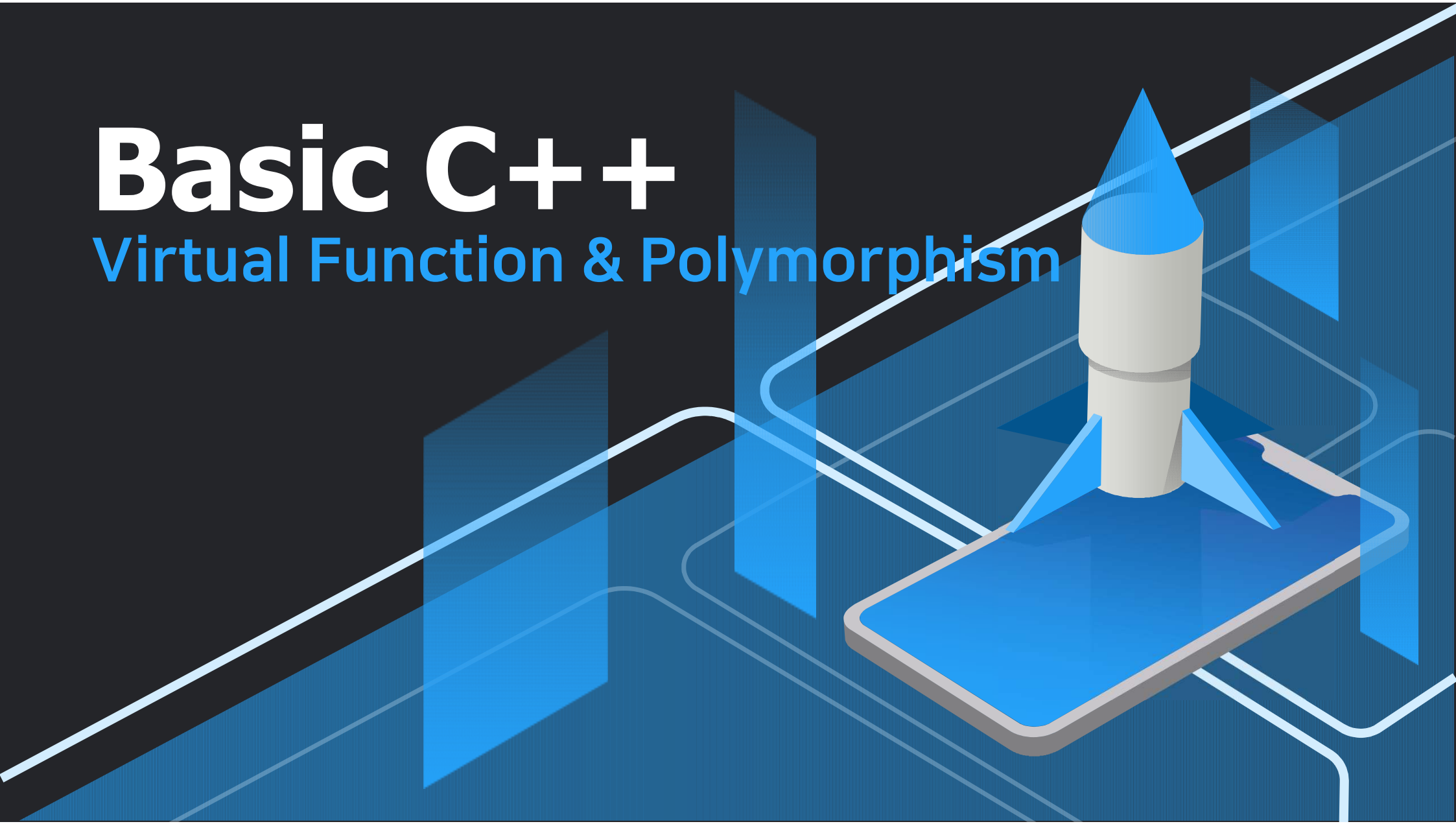


Basic C++

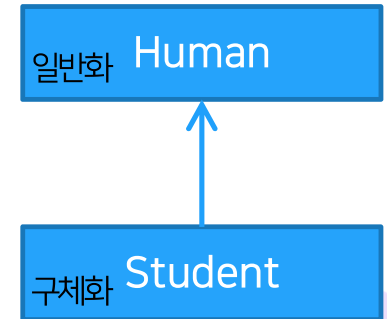
Virtual Function & Polymorphism



Basic C++ 목차

- Is-a / Has-a
- 오버라이딩 리뷰
- 오버라이딩 변경
- Virtual 키워드

Is-a / Has-a



1. 상속을 하는 이유

A. 실제 객체의 추상화를 더 효과적으로 하기 위한 것

```
class Student : public Human
```

B. 클래스 사이의 관계

- a. Student는 Human의 모든 기능을 포함
- b. Student는 Human의 기능을 모두 수행할 수 있기 때문에 Student 는 Human 이다라고 지칭 가능 (Student **Is a** Human)
- c. 자식(파생)클래스가 부모(기반)클래스를 가리킴
- d. Car는 Engine, Break, Tire, Audio 등을 가짐 / Product는 Date를 가짐 (Product **Has a** Date)

```
class Product
{
private:
    char name[64];
    char company[32];
    Date validto;
```

오버라이딩 리뷰

1. 소스 리뷰

```
class Human {
protected:
    char name[12];
    int age;
public:
    Human(const char* aname, int aage) { strcpy(name, aname); age = aage; }
    void intro() { printf("%s은 %d살 입니다\n", name, age); }
};

class Student : public Human {
protected:
    int stunum;
public:
    Student(const char* aname, int aage, int astunum) : Human(aaname, aage) { stunum = astunum; }
    void intro() { printf("%d번, %s입니다\n", stunum, name); }
};

int main() {
    Human h("성명건", 45);
    h.intro();
    Student s("최상식", 26, 25);
    s.intro();
}
```

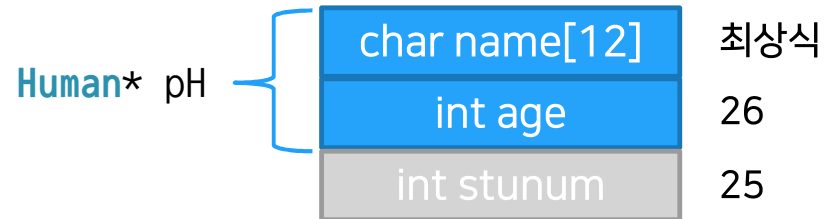
Microsoft Visual Studio 디버그 콘솔

성명건은 45살 입니다
25번, 최상식입니다

오버라이딩 변경

1. 소스 변경

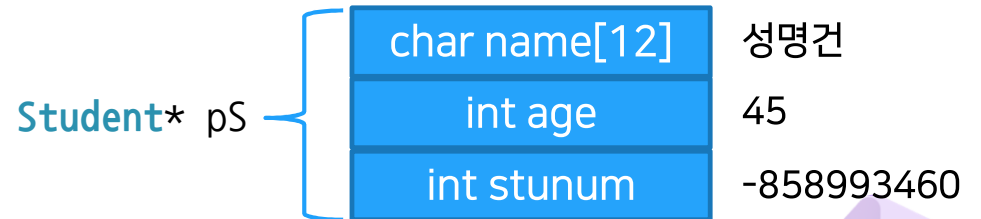
```
int main() {  
    Human h("성명건", 45);  
    Student s("최상식", 26, 25);  
    Human* pH;  
    Student* pS;  
  
    printf("Human size %d / Student size %d\n", sizeof(h), sizeof(s));  
  
    pH = &h;  
    pH->intro();  
    pS = &s;  
    pS->intro();  
    pH = &s;  
    pH->intro();  
}
```



Microsoft Visual Studio 디버그 콘솔

```
Human size 16 / Student size 20  
성명건은 45살 입니다  
25번, 최상식입니다  
최상식은 26살 입니다
```

오버라이딩 변경



1. 소스 변경

```
int main() {  
    Human h("성명건", 45);  
    Student s("최상식", 26, 25);  
    Human* pH;  
    Student* pS;  
  
    pS = &h; // Error  
}
```

2. 오류 수정 -> 타입 변환(캐스팅)

```
int main() {  
    Human h("성명건", 45);  
    Student s("최상식", 26, 25);  
    Human* pH;  
    Student* pS;  
  
    pS = (Student*) &h;  
    pS->intro();  
}
```

Microsoft Visual Studio 디버그 콘솔

-858993460번, 성명건입니다



Virtual 키워드

1. 다시 소스 변경!

```
class Human {
protected:
    char name[12];
    int age;
public:
    Human(const char* aname, int aage) {
        strcpy(name, aname);
        age = aage;
    }
    virtual void intro() {
        printf("%s은 %d살 입니다\n", name, age);
    }
};
...
int main() {
    Human h("성명건", 45);
    Student s("최상식", 26, 25);
    Human* pH;
    Student* pS;

    pH = &h;
    pH->intro();
    pH = &s;
    pH->intro();
}
```

Microsoft Visual Studio 디버그 콘솔

성명건은 45살 입니다
25번, 최상식입니다



Virtual 키워드

1. 설명

- A. 컴파일러 런타임 동안, pH의 시선
 - a. 난 Human 포인터니까 Human의 intro() 실행할래!
 - b. 음... intro는 virtual이군. 그럼 한 번 더 확인해야 겠다
 - c. 들어온 객체가... Human이구나, 맞네
 - d. Human의 intro 실행합니다
 - e. 다음!
 - f. 다시 머가 들어왔다... Student구나
 - g. intro가 virtual이었지? 그럼 Student에 있는 intro를 실행해야지



Virtual 키워드

1. 최종 변경

```
int main() {
    Human h("성명건", 45);
    Student s("최상식", 26, 25);
    Human* pH;
    Student* pS;

    int sel = 0;
    cout << "원하는 수를 고르세요(1:Human, 2:Student) ";
    cin >> sel;

    if (sel == 1)
        pH = &h;
    else if (sel == 2)
        pH = &s;
    else {
        cout << "맞는 수가 없습니다" << endl;
        exit(-1);
    }

    pH->intro();
}
```

Microsoft Visual Studio 디버그 콘솔

원하는 수를 고르세요(1:Human, 2:Student) 1
성명건은 45살 입니다

Microsoft Visual Studio 디버그 콘솔

원하는 수를 고르세요(1:Human, 2:Student) 2
25번, 최상식입니다



Virtual 키워드

1. 최종 변경 설명

- A. intro() 가 어느 객체의 것일 지 런타임시에 결정 : 동적 결합(Dynamic Binding) \leftrightarrow 정적 결합(Static Binding)
- B. 자식 클래스 멤버 함수와 부모 클래스 멤버 함수는 형태가 동일해야 하며, virtual 키워드가 붙은 함수를 가상 함수라고 함

2. const 키워드를 사용하면?

- A. 자식 클래스 intro()에 const 키워드 지정

```
class Student : public Human {
protected:
    int stunum;
public:
    Student(const char* aname, int aage, int astunum) : Human(aaname, aage) {
        stunum = astunum;
    }
    void intro() const { // override 사용
        printf("%d번, %s입니다\n", stunum, name);
    }
};
```

Microsoft Visual Studio 디버그 콘솔

성명건은 45살 입니다
최상식은 26살 입니다

결론

1. 정리

- A. Human 이냐 Student이냐에 따라 다르게 동작
- B. 하나의 함수(메서드)가 여러가지 다른 작업을 하는 것을 다형성(Polymorphism)이라고 칭함



수고 많으셨습니다

추가 문의사항이 있을 경우, 메일로 여쭙주시기 바랍니다.

personar95@naver.com

