گزارش تمارین

تمرین ۱

std::reference_wrapper<T> موجب ایجاد قالب استاندارد آدرس شیئ و یا تابعی از جنس T شود برای مثال
std::reference_wrapper<int> در واقع همان &int است و برای مواقعی است که نمی توان رفرنس شیئ یا تابعی را پاس داد و یا ذخیره کردن آدرس اشیاء در قالب استاندارد
و همچنین این قابلیت را میدهد که چند کپی از یک شیئ یا تابع با یک آدرس یکسان داشته باشیم
به قطعه کد زیر توجه کنید :

```
int m=10, n=20;
std::string s{"Hello"};

std::pair<int&, int> p1(m, n);
std::tuple<int&, int, std::string&> t1(m, n, s);
//Vs.
auto p2 = std::make_pair(std::ref(m), n);
auto t2 = std::make_tuple(std::ref(m), n, std::ref(s));
```

در این قطعه کد در دو خط آخر برای پاس دادن متغیر ها به تابع از reference wrapper استفاده شده زیرا ورودی ها از جنس آدرس هستند

دکتر خسروی تقریبا در تمام پاس دادن های با پوینتر شیئ از کلاس ها میتوانست از reference wrapper استفاده کند که ما در یکی از این کد ها این تغییر را ایجاد کردیم که موجب کمتر شدن خطوط کد و پیچیدگی کد شد و شاید کمی خوانایی بهتر

```
class C
{
 public:
    static reference_wrapper<C> obj()
    {
      return object;
    }

    int x=10;
    int y=20;

 private:
    static C object;
    C(){};
};
C C::object= C();

int main() {
    C& a=C::obj();
    C& b=C::obj();

    cout << &a << endl<<&b<<endl;
    a.x=a.x+b.y;
    cout << a.x << endl<<b.x<<endl;

    return (0);

}
```

مدت زمان انجام این تمرین ۲ ساعت
مدت زمان انجام تمرین ۳ : ۴ ساعت