

1 Declarations and Definitions from §6.3

```
char ch;
string s;
auto count = 1;
const double pi {3.1415926535897};
extern int error_number;
const char name = "Njal";
const char season[] = { "spring", "summer", "fall", "winter" };
vector<string> people { name, "Skarphedin", "Gunnar" };
struct Date { int d, m, y; };
int day(Date p) { return p>d; }
double sqrt(double);
template<class T> T abs(T a) { return a<0 ? -a : a; }
constexpr int fac(int n) { return (n<2)?1:nfac(n1); }
constexpr double zz { iifac(7) };
using Cmplx = std::complex<double>;
struct User;
enum class Beer { Carlsberg, Tuborg, Thor };
namespace NS { int a; }
```

2 Definitions

(local scope [class / function] assumed)

```
char ch = 'a';
string s = {};
int error_number = 0;
double sqrt(double z) { /* Newton Iteration Algorithm */ }
struct User { string name; int age; };
```

3 Declarations

(local scope [class / function] assumed)

```
extern const double pi;
extern const char* name;
extern const char* season[];
vector<string> people;
extern struct Date;
int day(Date* p);
template<class T> T abs(T a);
extern constexpr int fac(int n);
extern constexpr double zz;
extern enum class Beer;
namespace NS { extern int a; }
```