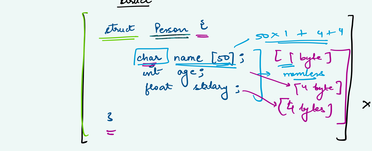
Structures

→ Collection of variables of different data types under a single name.

struct



→ struct definition is a blueprint for creating variables.

→ Person Ajay; → 58 bytes

Ajay.age = 50;

→ Pointers can also be created for user defined data-type like structure.

```
struct temp {
    int i;
    char g;
};

int main() {
    temp *ip;
    return 0;
}
```

Distance *ip, d;

ip = &d;

(*ip).inch → higher precedence

(*ip)→inch

d.inch

ptr → feet = (*ptr).feet

Enumeration (enums)

An enumeration (enum) is a user-defined data-type that consists of integral constants.

enum

enum season { spring, summer, autumn, winter }

spring = 0, summer = 1, autumn = 2, winter = 3

```
enum season {
    spring = 0,
    summer = 4,
    autumn = 8,
    winter = 12
};
```

→ blueprint of the variable is created

enum season current;

① enum boolean { false, true };

enum boolean check;

② enum boolean {

false,

true

} check;

=

```
enum cards {
    club = 0;
    diamonds = 10;
    hearts = 20;
    spades = 3
};
```

Classes & Objects

className VariableName;

```
int main() {
    Room r1, r2, r3;
    r1.area();
    r1.length = 20.2;
}
```

Constructors

① A constructor is a special type of member function that is called automatically when an object is created.

```
class Wall {
public:
    Wall() {
    }
};

Wall w;
```

default constructor

constructor

Parameterized constructor

Room (double len, double b, double w) {

}

Copy constructors

→ used to copy data of one object to another.

int *p ✓

int &p ✓

int &p; p ✗✗