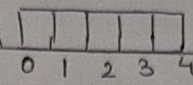
- Array
- Structure

Date 25/11/2021
Page 1

C and C++ Concept

→ **Arrays:-** list of same data type in consecutive memory location.

int A[5];

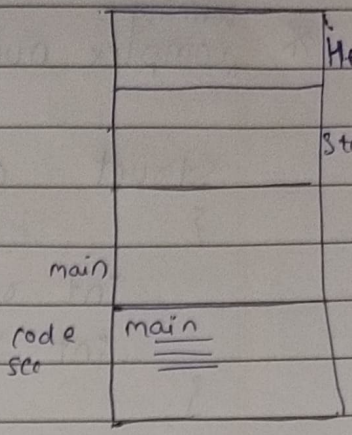


B[5] = { 1, 9, 8, 7, 0 }

↑ declaration of array

↳ Initialization of array

Main Memory



→ a[5] = {0}

printf (a[2]) → 0

because a[0] was initialized

→ for(int x:A) → for each element as x in array A

{

x = x * 2

print(x)

}

→ **Structure :-** Collection of related data member (can be same or different data type) in one name

- Struct Rectangle → as rectangle has 2 member length & breath

{

int length; → 2 byte or 4 byte } in Struct will have 2 or 8 byte

int breath; → 2 or 4 byte

};

int main()

{

struct Rectangle r; ← declaration

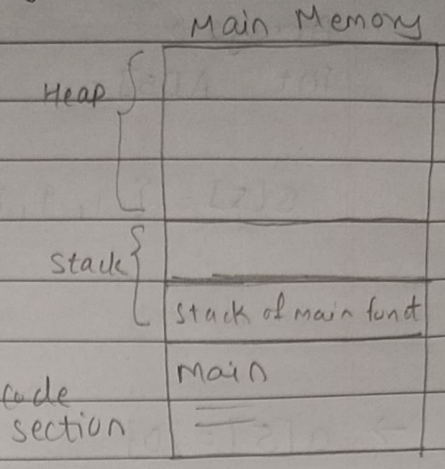
struct Rectangle r = {10, 5} ← declaration & initialization

- Structure.
- Basics of Cards.

Date 25/11/2024
Page

for accessing member of structure is used. . (dot) operator

r.length = 15;



* Defining complex number in structure.

```
struct complex
{
    int real;
    int img;
};
```

* Cards

Face → Axe 1, 2, 3 ···· 10, J, K, Q

Shape → ◇, ♡, ♧, ♤
0 1 Club 2 spade 3

color → Black, Red
0 1

```
Struct card
{
    int face;
    int shape;
    int color;
};

int main()
{
    struct card c;
    c.face = 1;
    c.shape = 0;
    c.color = 0;
```

struct card c={1,0,0}

"%lu" → for long unsigned int

"%d" → for int

Date
Page 25/11/2021

```
int main()
{ struct card deck[52];

  for (i=0; i<52; i++)
  {
     cout << "face";
     cin >> deck[i].face; cout << endl;
     cout << "Shape";
     cin >> deck[i].face;
     cout << endl << "color";
     cin >> deck[i].color;
  }

  for (i=0; i<52; i++)
  {
     cout << deck[i].face << " " << deck[i].shape << " " << deck[i].
color;
```