

struct tagOfStructure

{

Definition of Information Structure

};

struct MassAndDistance

{ double m1;  
double m2;  
double r;  
};

struct Date

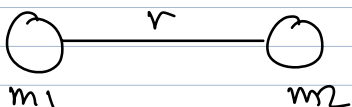
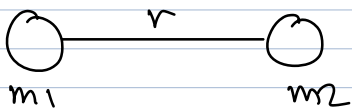
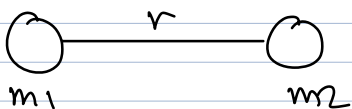
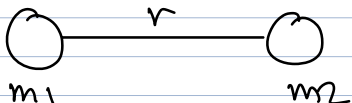
{

};

struct DebitCard

{

};



struct MassAndDistance

{

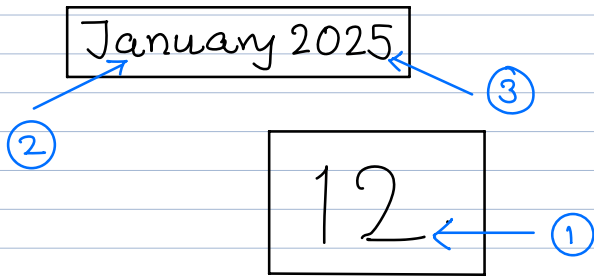
double m1;

double m2;

double r;

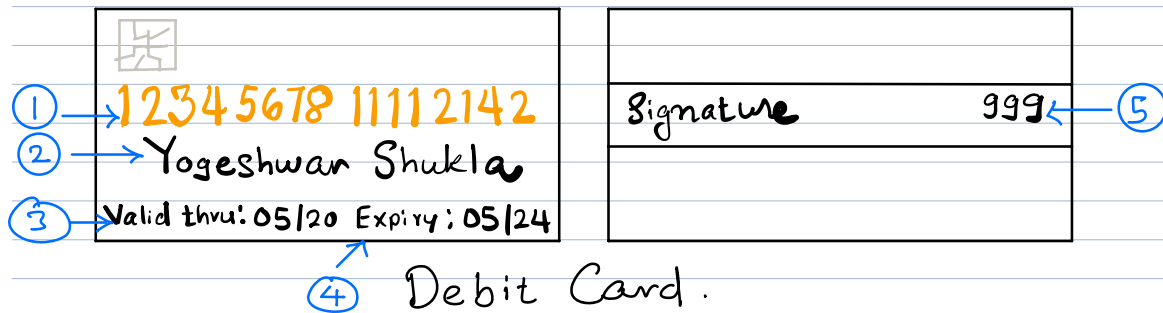
};

Calendar



Struct Date

```
{  
    int day;  
    int month;  
    int year;  
};
```

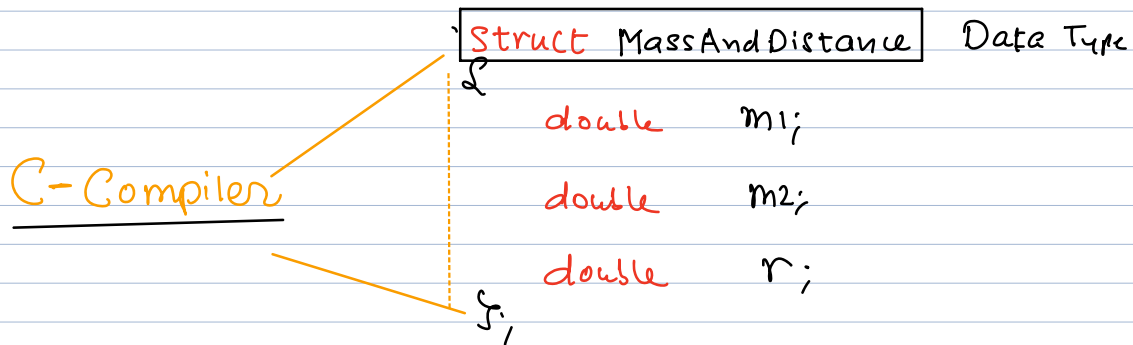


Struct DebitCard

```
{  
    unsigned long long dc_card_number;  
    char dc_holder_name[128];  
    Struct Date dc_valid_thru_date;  
    Struct Date dc_expiry_date;  
    int dc_cv;  
};
```

Short int long long long

2 4 4/8 8



Compiler will not allocate a single byte of memory at this point. It will simply record the information structure to itself for future purposes.

int n;

Struct MassAndDistance earthSun;

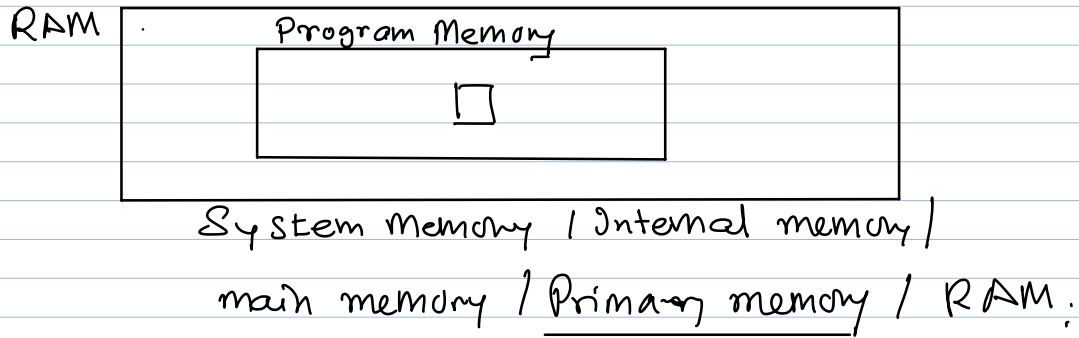
<div> <div>Struct MassAndDistance</div> <div> double m1; double m2; double r; §, </div> </div>	<div>Struct MassAndDistance earthSun;</div>
--	---

# compiler source code  
exe file.

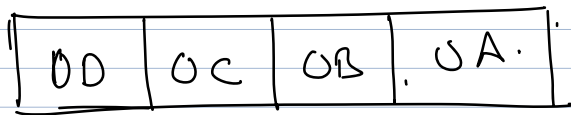
# exe file → run

Executable file run करवाकर  
याची process चलाय होते.

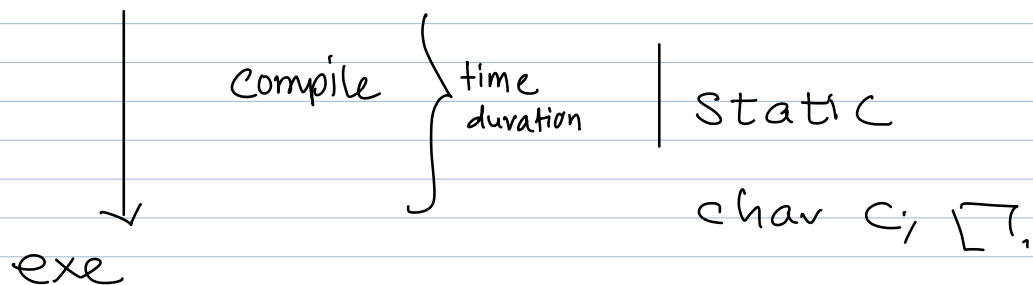
Running Instance of  
executable file is a process.



```
int num = 0x0A0B0C0D;
```



Source code लिखते हैं

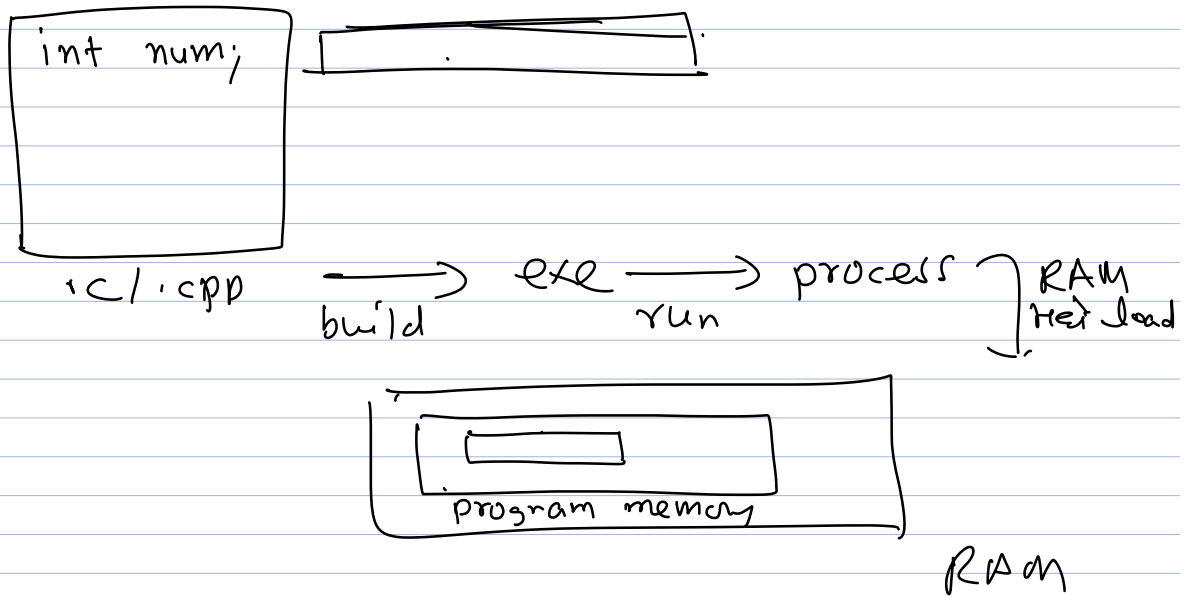


exe → launch

Process → start

Process → end

time | dynamic run-time



## Source Code

### GLOBAL SCOPE

```
int main(void)
{
```

LOCAL SCOPE of main()

```
}
```

```
void sort(int* p, int N)
```

```
{
```

LOCAL SCOPE of sort()

```
}
```

```
void test_sort(int* p, int N)
```

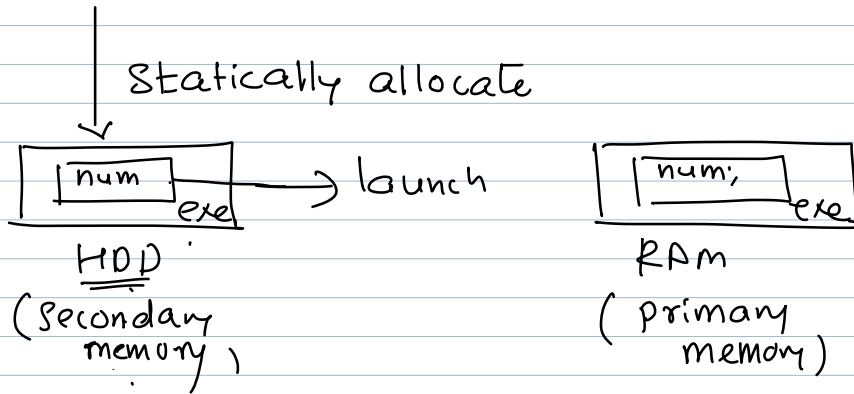
```
{
```

LOCAL SCOPE of test\_sort()

```
}
```

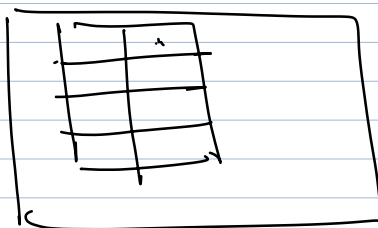
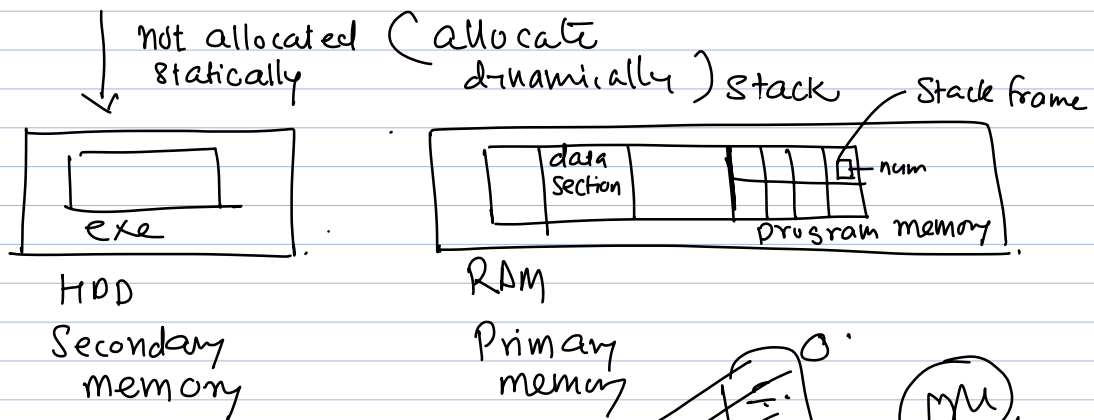
int num; // Data Definition Statement

Global scope



int num;

local scope



Graph

