Date: / /20 M T W T F S

OOP (Week 1)

- Migration from modular program having structures and functions having classes & Objects

↳ Procedural programming is based on creating procedures & functions

↳ In procedural programming, data and functions are separate entities.

↳ Objects oriented programming is based on creating objects.

⇒ • Classes provides encapsulation of member variable and member functions.

- Classes provide data hiding from outside world.
- Classes provide reusability of the code (saves time & effort).
- Procedural programming makes program more complex while object oriented programming makes program simple and easily understandable.

Syntax and Semantics

```
Class Class_Name{
        //member variables
Public:
        //member functions.
};
```

- By default the data of the class is private.
- We can use Private & Public keyword in classes multiple time

e.g

```
class Class_Name{
Public:
        //member function.
Private:
        //member variable.
Public:
        //member function
};
```

- It is a convention (not compulsory but good practice) to write

~~d~~ 1st letter of class name as a capital letter.

Implicity available member functions:

These functions are available in the class (by default), if we don't write them.

i) Default Constructor.

ii) Copy Constructor.

iii) Assignment Operator. (=)

iv) Address-of operator. (&)

Default Constructor:

for declaring the member variables of the object.

Copy Constructor:

for copying, member variables of one object into another.

Assignment Operator:

for copying, member variables of one object into another (after declaration of object).

Address-of operator (&)

↳ Gives the address of current object.

Access Modifiers:

i) Pubic:

↳ To allow the outside world to access member functions & member variable.

ii) Private:

↳ To hide ~~da~~ member variable & member functions from outside world.