Day 5

- class is keyword in java.
- Using class we can group related data elements.
- data element declared inside class is called field.

```
class Complex
{
    int real;   //Field
    int imag;   //Field
}
```

- Process of creating object/instance from class is called instantiation.
- If we want to create object of a class / instantiate class then we should use new operator.
- In C Instantiation:

```
struct Complex c1;
struct Complex *ptr =  ( Complex* )malloc( sizeof( Complex ) );
```

- In C++ Instantiation:

```
class Complex c1;   //Ok
Complex c2;   //Ok
Complex *ptr = new Complex();   //OK
```

- If we allocate memory using new operator then it gets reserved on heap section. Everything on heap section is annonymous.
- In Java

```
new Complex( ); //Annonymous Instance
```

- If we want to perform opeartions on instance then we should create object reference / reference.

```
//Complex c1; //It is object in C++.
Complex c1; //It is object reference in java.
new Complex( ); //Annonymous Instance
Complex c2 = new Complex();
```

- If we create instance without reference then it is called anonymous instance.

```
int number = 10;    //Initialization
```

- During declaration of variable, we strore value inside it then it is called initialization.
- In the lifetime of the variable/instance we can initialize that variable only once.
- If we want to process state/value of the instance then we should call method on it.
- Function defined/implemented inside class is called method.

```
    public static void main(String[] args)
    {
       Complex c1 = new Complex( );
       c1.printRecord( ); //Message Passing
    }
```

- In above code, printRecord() method is called on c1 object(actually c1 is reference).
- Process of calling method on instance is called message passing.

this reference

- If we want to process state/value of the instance then we should call method on it.
- If we call method on instance then compiler(javac) implicitly pass reference of current instance as a argument.

```
c1.printRecord();   //c1.printRecord( c1 );
```

- To store reference of current instance, compiler implicitly declare one reference as a parameter. Such parameter is called this reference.

```
class Complex
{
    void printRecord(/*Complex this*/)
    {   }
}
```

- this is a keyword in java.
- Using this reference, field and method can communicate with each other hence this is called connection/link between them.
- "this" is implicit reference variable taht is available in every non static method of the class which is used to store reference of current instance or calling instance.

Constrcutor

- It is a method of a class which is used to intialize field/instance.
- Constructor is special because:

    1. Its name is same as class name.
    2. It doesn't have any return type.
    3. It is designed to call implictly.
    4. It gets called once per instance.
- Types of constructor:
    1. Parameterless / zero argument / User Defined default constructor
        - A constructor which do not take any parameter is called parametrless constructor.
        - If we create instance without argument then parameterless constructor gets called.
        - Complex c1 = new Complex(); Here on instance parameterless constructor will call.
    2. Parameterized constructor
        - A constructor which take parameters is called parameterized constructor.
        - If we create instance with argument then parameterized constructor gets called.
        - Complex c1 = new Complex(10,20); Here on instance parameterized constructor will call.
    3. Default Constructor(Compiler Supplied)
        - If we do not define constructor inside class then compiler generates default constructor for the class.
        - Default constructor is zero argument constructor. Compler never provides default parameterized constructor.
- We can define multiple constructors inside class. It is called constructor overloading.
- In Java, we can call constructor from another constructor. It is called constructor chaining.
- For constructor chaining we should use this statement,
- this statement must be first statement inside constructor body.
- If we want to reduce developers effort then we should use constructor chaining.

Flow to write and use class:

0. Understand problem statement.
1. Write Empty class.
2. Depending on the requirement, declare field(s) inside class.
3. Instantiate class.
4. Define constructor to initialize instance
5. To process state of instance call method on it.
6. Using this process state inside method.

- Fields get space per instance.
- Method do not get space inside instance. Rather all the instances of same class share single copy of it.(Using this reference instance can share method)

Characteristics of instance

1. State
    - Value stored inside instance is called state.
    - Value of the field represent state of instance
2. Behaviour
    - Operation that we can perform on instance is called behavior.
    - Method of a class represents behavior of instance
3. Identity
    - Value of any field that is used to identify instance uniqely is called identity.