1 JAVA ORIENTATION

Section- 1

- Any problem, any logic, for that matter, anything can or rather has to be put inside a CLASS.in JAVA.

- So fill up tour code in a class and save it as class name.java(*preferably*)
  Example:

```
class ABC
    {
        int a;
        String s;
        void a()
            {
                System.out.println("amethod");
            }
    }
```

Class name for the above program is:**ABC.java** (*preferable but not compulsary*)

Note:But if you make a class public, you must name that file with that class name.

Example:

```
public class ABC
    {
        int a;
        String s;
        void a()
            {
                System.out.println("amethod");
            }
    }
```

Class name is : **ABC.java** (*MANDATORY*)
CDE.java (*COMPILATION ERROR*)

- A .java file can contain more than one class but only public class. **Example:**

```
class ABC
    {
        int i;
        String s;
        void a()
            {
                System.out.println("HI");
            }
    }
class B
    {
        char c;
        void b()
            {
                System.out.println("HELLO");
            }
```

```
}
```

Class name for the above program can be : **A.java**
(*or*)
B.java
(*or*)
some random name.java

But, The following program will fail.

Example:

```
public class ABC
    {
        int i;
        String s;
        void a()
            {
                System.out.println("HI");
            }
    }                   public class B
    {
        char c;
        void b()
            {
                System.out.println("HELLO");
            }
    }
```

Class name for the above program: **A.java**
(*or*)
B.java

THESE TWO CLASS NAMES ARE WRONG.

Reason : A .java should not contain more than one public class.So, the correct version is the following:

Example:

```
public class A
    {
        int i;
        String s;
        void a()
            {
                System.out.println("HI");
            }
    }
class B
    {
        char c;
        void b()
            {
                System.out.println("HELLO");
            }
    }
```

Class name for the above program: **A.java** (*COMPULSORY*)

EXAMPLES OF OTHER CLASSES

eg:1

```
class Test
{
    public static void main(String[]args)
    {
        System.out.println("Helloworld");
    }
}
```

Class name is **Test.java**

eg:2

```
class Student
{
    int rollno;
    String name;
    String degree;
    void read()
    {
        System.out.println("Reading");
    }
    void write()
    {
        System.out.println("writing");
    }
}
```

Section-2

Any .java has to pass through two phases:

1.Compilation Phase
2.RunTme Phase

1.Compilation Phase:

Any java program is compiled using "javac filename.java"

Example:1

```
class ABC
{
    int i;
    void a()
    {
        System.out.println("HI");
    }
}
```

ABC.java(*classname*)$\longrightarrow$javac ABC.java(*compiling*)$\longrightarrow$ABC.class

After Compilation classname.class will be produced.

Example:2

```
public class ABC
{
```

```
        int i;
        void b()
            {
                System.out.println("HI");
            }
    }
class B
    {
        B()
            {
                System.out.println("B");
            }
    }
class C
    {
        void d()
            {
                System.out.println("d");
            }
    }
```

Class name for the above program: **ABC.java** and after compiling the above program using javac ABC.java three .classes will be produced i.e. **ABC.class, B.class, c.class**.

What happens in Compilation error

Compliler will check for the following:

1.Syntax Checking

```
//A.java
class A
    {
        int i⟶(compilation falis because no semicolon;)
int j;
        void a();⟶(Complilation falis because semicolon should not be placed during defination of method)
        {
            System.out.println("d");
        }
    }
```

2.Wrong Assignments

```
//A.java
eg:1    class A
        {
            public static void main(String[]args)
            {
                float f = 19.1;⟶(Compilation falis because assigning double(19.1) to float f will cause of precision)
            }
        }
```

eg:2 More common case is with "reference" variables.

```
//Cat.java
                class Cat
                    {
                        int size;
                        int height;
                        void talk()
                            {
                                System.out.println("meow");
                            }
                    }
// Dog.java
                class Dog
                            {
                        int size;
                        int height;
                        void talk()
                            {
                                System.out.println("bow");
                            }
                    }
//Test.java
                    {
                        public static void main(String[]args)
                            {
                                Cat c= new Cat();
                                Dog d= new Dog();
                                d=c;------>(Compilation will fail because as-
sign "Cat" variable to "Dog" will not be allowed)
                            }
                    }
```