

Subject → 'OOAD'

Assignment - 1

Q1 Explain different types of object oriented features along with an example.

Ans.) There are different features of such as:-

(i) Object :- Objects are the basic runtime entity. They may represent a person, place, account etc. It is the instance of class and consist of method and properties to make a particular type of data useful.

eg:-

```
class person
```

```
{
```

```
    functions
```

```
    // statement
```

```
}
```

```
int main()
```

```
{
```

```
    person p1; // p1 is an object
```

```
}
```

(ii) Class :- It is a user define datatype which hold its own data member & member function which can be accessed by an object of that particular class. It is blueprint for object.

eg:-

```
class person
```

```
{
```

```
    void func1()
```

```
{
```

```
    // body of function
```

```
}
```

```
    void func2()
```

```
{
```

```
    // func 2
```

```
}
```

```
}
```

∴ Here, a class (person) consist of its own function.

3. Encapsulation:- Wrapping up of all data and object under a single unit is called Encapsulation or we can say that containing data & funcⁿ in unit called class leads to encapsulation.

Consider a real life eg of encapsulation, in a company, which contain different section like accounts & sales etc. Accounts of company and sales section. They have to contact same other officer and to give particular data. This is what encapsulation is. Here data of sales section & its particular employees are wrapped only in a single name "sales section".

4. Data Abstraction:- It means displaying only essential information and hiding the details. That is providing essential info. about data to outside world, hiding the background details or info.

eg:- In an ATM a person can only retrieve money when ever he want but he has no idea how the transaction from his account is actually done or one can say that he can use ATM but don't know its implementation.

5) Poly morphism:- It means a state of having many shapes or ability to take diff. forms. Real life eg can be seen as a woman at the same time can have diff. characteristic like, daughter, mother, employer etc. It is of 2 types operator over loading & func. over loading

eg:- class Multiplication

```
{
    static int Multiply (int a, int b)
```

```
{
    return a*b;
```

```
}
```

```
}
```



```

static double Multiply (double a, double b)
{
    return a * b;
}

```

```

class Main { public static void main (String[] args) {
    System.out.println (Multiply fun. Multiply (2, 4));
    System.out.println (Multiply fun. Multiply (5.5, 6.3));
}
}

```

3

O/p 8

34.65

6. Inheritance :- A capability of a class to derive properties & characteristics from ^{other} class

eg:- class Parent

```

{
    public:
    int id.p;
}

```

3;

class child: public parent

```

{
    public:
    int id.c;
}

```

3;

int main ()

```

{
    child c;
    c.id.c = 7;
    c.id.p = 9;
}

```

cout << "child id = " << c.id.c;

cout << "parent id = " << c.id.p;

Φ :-
n
q1

7. Dynamic Binding: - Refers to linking of a function call with function def. is called binding and when it takes place at runtime is called dynamic binding.

8. Message passing: - The process by which one object can interact with other object is called message passing.

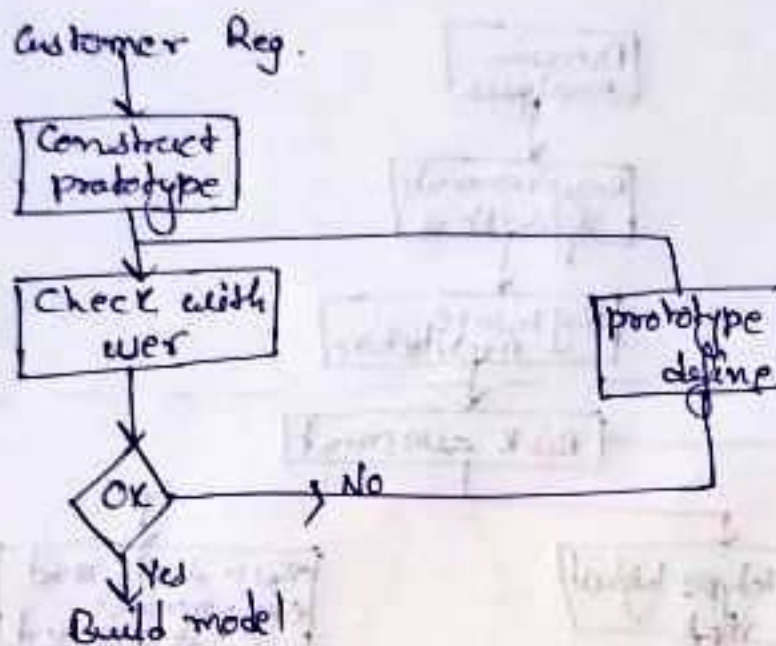
Q2 > what is meant by iterations. Explain iterative development model.

Ans > Iterations allow us to focus on a subset of the complete project as we work out the details.

- We discover new problems and requirement during the process of building one of the subsystem.
 - The process allow us to test each subsystem independency and assure its proper functionality.
 - When reach the final stage of development, integration of subsystem into a whole, we focus on integration that each subsystem has been fully tested.
 - Prototype model is an example of iterative development.
- Iterative development.

The software process or software development associated with the object oriented development is inherited different from traditional software life cycle.

Example :-



The Iterative model :-

It allows the accessing earlier phase, in which the variations made respectively. The final o/p of project renewed at end of SDLC process

Iteration 1

→ Requirement

→ Analysis

→ Design

→ Testing

→ Implementation

→ Review

Iteration 2

Design

Testing

Implementation

Review

Iteration 3

Design

Testing

Implementation

Review

Deployment
Maintenance

ITERATIVE MODEL

Example : -

