welcome

VIVEKANAND COLLEGE, KOLHAPUR

(AUTONOMOUS)

DEPARTMENT OF BCA

Basic Concepts of OOP

Presented By: Mr. Vijay Bapuso Pujari

CONTENT

Introduction

Basic concepts of OOP'S

INTRODUCTION TO C++

- C++ is a high level “semi-object oriented” programming language ,developed by bjarne stroustrup in 1979 at bell labs.

- C++ is the next version of c language.

BASIC CONCEPTS OF OOPS

1.CLASS

2.OBJECT

3.ABSTRACTION

4.INHERITENCE

5.POLYMORPHISM

6.ENCAPSULATION

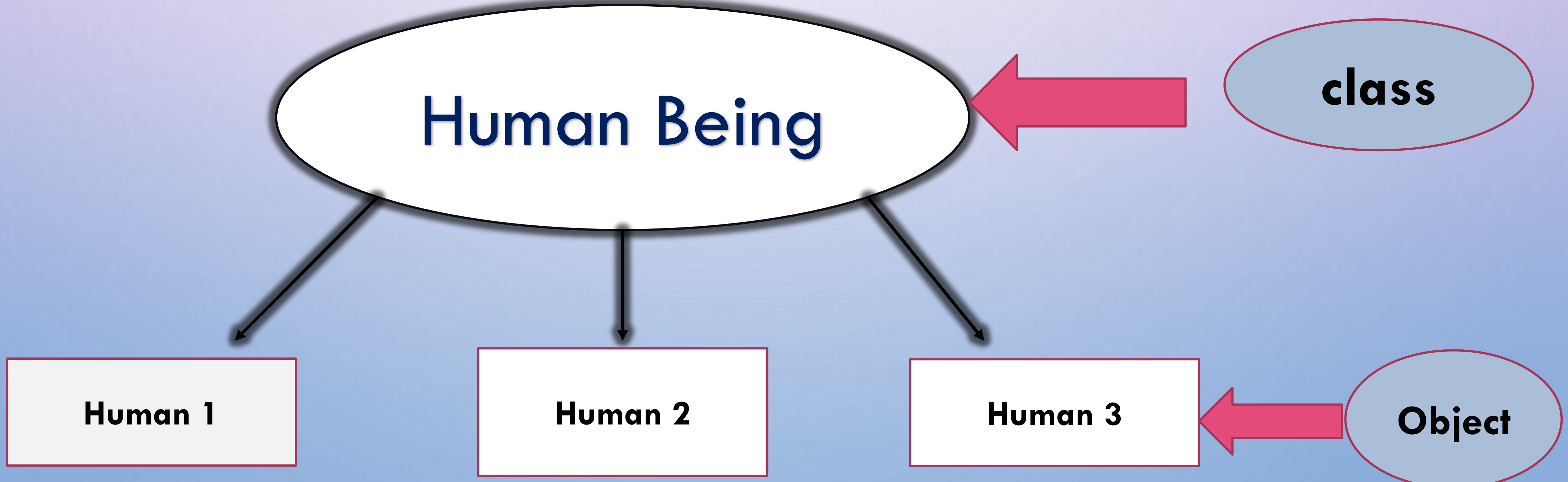
7.MESSAGE PASSING

CLASS

- Class is a collection of object.
- Class is a passive entity.
- A class is user defined data type which has data member and member function.

OBJECT

- Any entity that has state and behavior is known as object.
- Object is a active entity.



INHERITANCE

- ❖When one class access the property of another class is called **inheritance**.
- ❖Inheritance is the process in which there is a **relationship** between two base class and **derived** class.

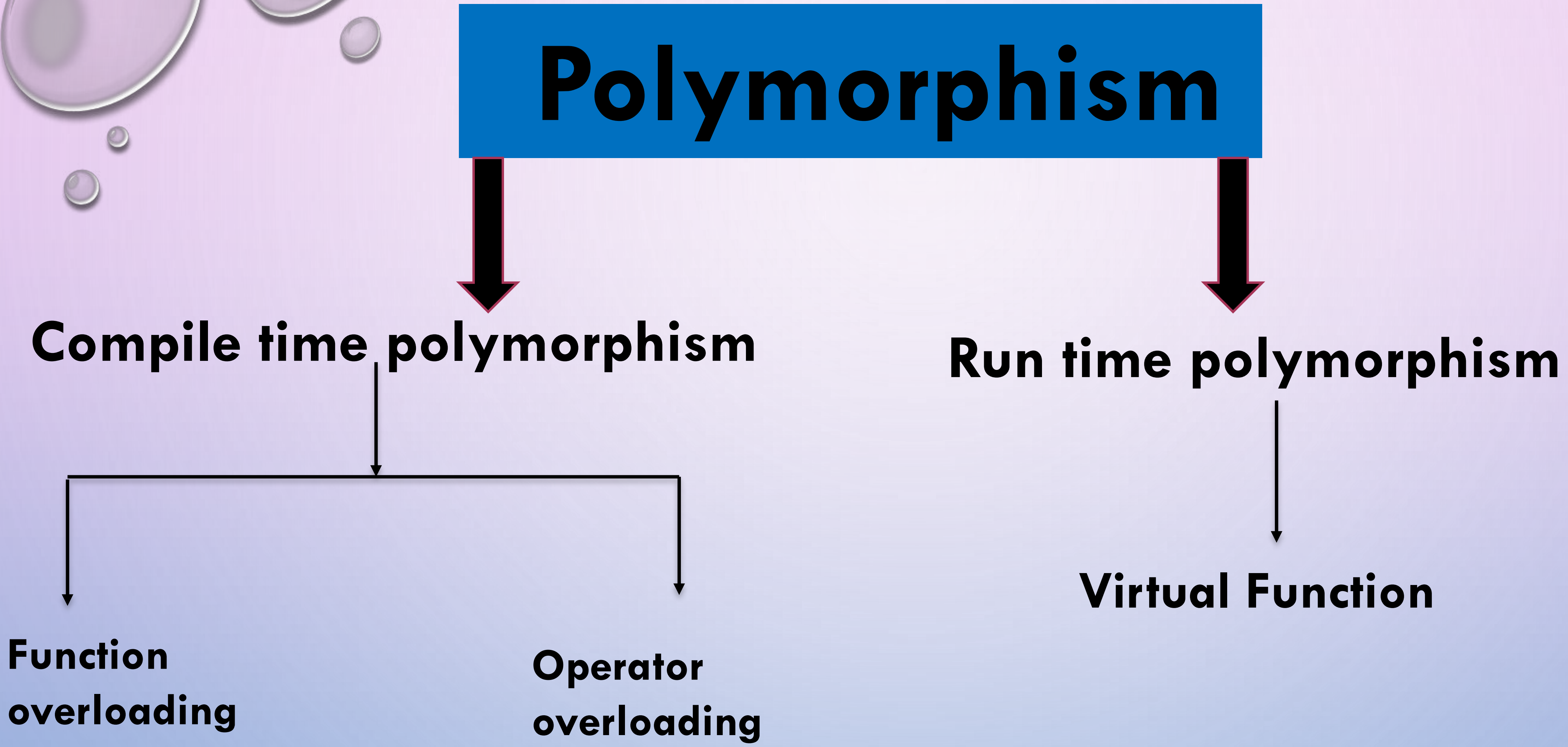
POLYMORPHISM

❖When one task is performed by different ways are known as polymorphism.

For e.g

Polymorphism
Sitesbay.com
Sitesbay.com

A person at the same time can have different character like a father, a husband, an employee so the same person passes different behavior this is call polymorphism.



Polymorphism

Compile time polymorphism

Function overloading

Operator overloading

Run time polymorphism

Virtual Function

ABSTRACTION

- Hiding internal details and showing functionality.

For e.g phone call, we don’t know the internal processing.

ENCAPSULATION

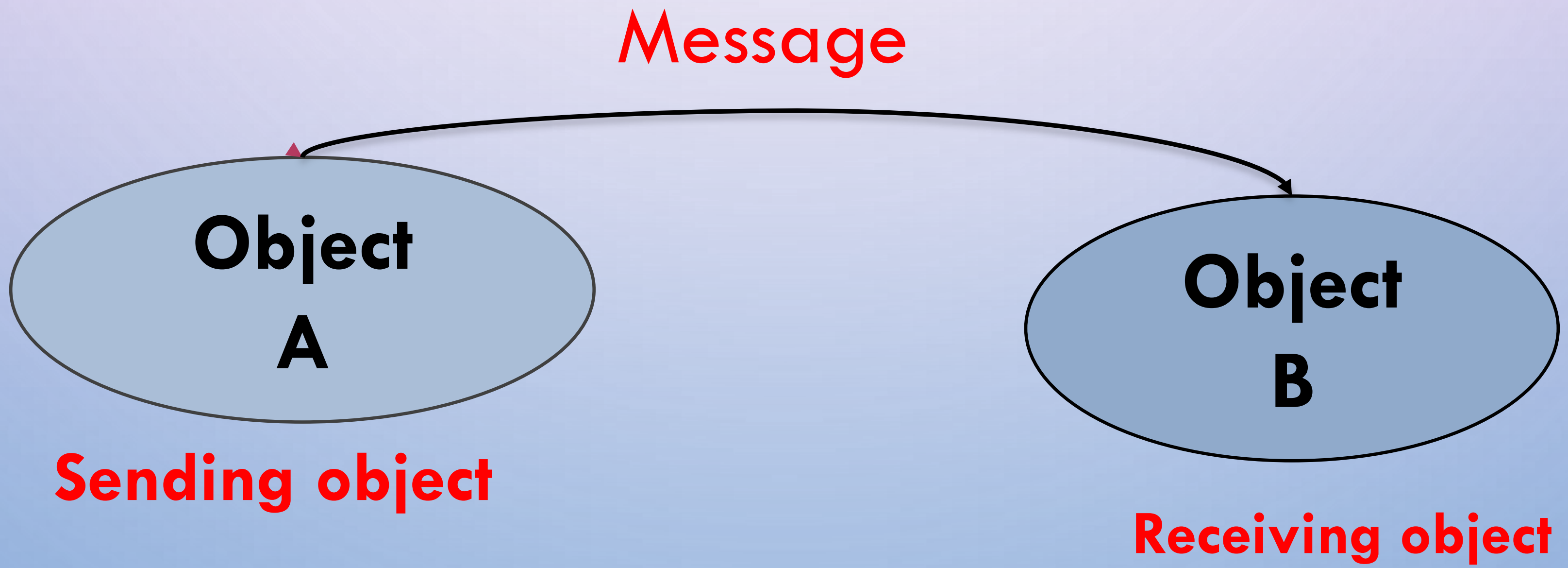
- Binding of code and data in a single unit is know as encapsulation.
- Example capsule, it is wrapped with Different medicines.

DYNAMIC BINDING

- In dynamic binding the code to be executed in response to function call decided at runtime.
- C++ has virtual function to support this.

MESSAGE PASSING

❖ Object communicate with one another by sending & receiving information to each other through message passing.



Message

Object A

Sending object

Object B

Receiving object

Thank you!