Object-oriented programming (OOP) is a key concept that involves message passing between objects. In OOP, objects communicate by sending messages to each other requesting for services or information. The idea of message passing in achieving encapsulation, modularity and the separation of concerns is at the core of this concept.

An outline of message passing in OOP is provided below:

1. Object Interaction:

- In OOP, systems are modeled as interacting objects, each responsible for a specific aspect of the overall functionality.
- Objects interact with each other by sending messages.

2. Messages:

- A message is a request for an object to perform one of its methods (functions).
- It represents the communication between objects and typically includes the name of the method to be executed and any required parameters.

3. Methods:

- Methods are functions associated with a class. They define the behavior of objects of that class.
- When an object receives a message, it invokes the appropriate method to respond to the request.

4. Encapsulation:

- Message passing facilitates encapsulation by allowing objects to hide their internal state and expose only the necessary functionality through their methods.
- Objects can interact with each other without directly accessing each other's data.

Message passing example in C++ language:

C++

```cpp
// Class definition
class MessageReceiver {
public:
    // Method to process a message
    void processMessage(const std::string& message) {
        std::cout << "Message received: " << message << std::endl;
    }
};

int main() {
    // Create an object of MessageReceiver
    MessageReceiver receiver;

    // Send a message to the object
    receiver.processMessage("Hello, Object!");

    return 0;
}
```

In this example, the MessageReceiver class has a method processMessage that can be called to process a message. The main function creates an object of MessageReceiver and sends a message to it.

Related posts:

1. Abstraction and encapsulation
2. Object Oriented Programming & Methodolog Viva Voce
3. How to install compiler for code blocks
4. Object Oriented Programming
5. Differences between Procedural and Object Oriented Programming
6. Features of Object Oriented Paradigm
7. Inheritance in Object Oriented Programming
8. Object Oriented Programming
9. Introduction to Object Oriented Thinking & Object Oriented Programming
10. Difference Between Object-Oriented Programming (OOP) and Procedural Programming
11. features of Object oriented paradigm
12. Merits and demerits of Object Oriented methodology
13. Concept of Objects: State, Behavior & Identity of an object
14. Access modifiers
15. Static members of a Class
16. Instances in OOP
17. Construction and destruction of Objects