

```
        house.setRoof("Wood, caribou and seal skins");
    }

    public House getHouse()
    {
        return this.house;
    }
}

class CivilEngineer
{
    private HouseBuilder houseBuilder;

    public CivilEngineer(HouseBuilder houseBuilder)
    {
        this.houseBuilder = houseBuilder;
    }

    public House getHouse()
    {
        return this.houseBuilder.getHouse();
    }

    public void constructHouse()
    {
        this.houseBuilder.buildBasement();
        this.houseBuilder.buildStructure();
        this.houseBuilder.bulidRoof();
        this.houseBuilder.buildInterior();
    }
}

class Builder
{
    public static void main(String[] args)
    {
        HouseBuilder iglooBuilder = new IglooHouseBuilder();
        CivilEngineer engineer = new CivilEngineer(iglooBuilder);

        engineer.constructHouse();

        House house = engineer.getHouse();

        System.out.println("Builder constructed: "+ house);
    }
}
```

[Run on IDE](#)

Output :

```
Builder constructed: House@6d06d69c
```

Advantages of Builder Design Pattern

- The parameters to the constructor are reduced and are provided in highly readable method calls.
- Builder design pattern also helps in minimizing the number of parameters in constructor and thus there is no need to pass in null for optional parameters to the constructor.
- Object is always instantiated in a complete state
- Immutable objects can be build without much complex logic in object building process.



Disadvantages of Builder Design Pattern

- The number of lines of code increase at least to double in builder pattern, but the effort pays off in terms of design flexibility and much more readable code.
- Requires creating a separate ConcreteBuilder for each different type of Product.

This article is contributed by **Saket Kumar**. If you like GeeksforGeeks and would like to contribute, you can also write an article using contribute.geeksforgeeks.org or mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.

GATE CS Corner Company Wise Coding Practice

Design Pattern

Recommended Posts:

[Design Patterns | Set 1 \(Introduction\)](#)

[Bridge Design Pattern](#)

[Adapter Pattern](#)

[Template Method Design Pattern](#)

[How to design a parking lot using object-oriented principles?](#)

[Data Access Object Pattern](#)

[Front Controller Design Pattern](#)

[Business Delegate Pattern](#)

[MVC Design Pattern](#)

[Intercepting Filter Pattern](#)

(Login to Rate and Mark)

0

Average Difficulty : **0/5.0**
No votes yet.

Add to TODO List

Mark as DONE

Writing code in comment? Please use ide.geeksforgeeks.org, generate link and share the link here

Load Comments

Share this post!