Designe patterns:

A software designe pattren is a genral, reusable solution to a commonly occuring problem. It is not a finished designe that can be directly transformed into source or machine code. It is a description or template for how to solve a problem that can be used in many different situations.

Builder:

Builder is a creational designe pattern that lets u construts complex objects step by steps. The pattern allows you to produce different types and representation of object using the same construction code.

Problem:

Imagine a complex problem/object that requires laborious, step by step initilization of many feild and nested objects.
For example lets think about how to create a house object. To build a simple house you need to construct four walls and a flour, install a door, fit a pair of window and build a roof.
But if you want a bigger, brighter house with back yard and other goodies (like heating system, plumbing and electrical wiring) -

Date:..................

The simplest solution is to extend the base house class and create a set of subclases to cover all combination of the parameters. It will end up with a considerable number of subclasses. Any new parameter will require growing this hierarchy even more.

Solution:
The builder pattern suggests that you extract the object construction code out of its own class and move it seprate objects ~~class~~ called builders.

The builder pattern lets you constructs complex objects step by step. The builder dosent allow the other objects to access the product while it being built.
The important part is that you don't need to call all of the steps. You can call only those steps that are necessry for producing a particular configuration of an object.

Some of the construction steps might require different steps for implimentation when you need to build various representation of one of the product.

For example:
wall for cabins may be built of wood but castle wall may be build with stone

ALBA

In this case you can create several different builder classes that implement the same set of building steps, but in different manner

Imagine a builder that build everything from wood and glass and a second that build everything with stone and iron and third one that uses gold and diamond. By calling the same set of steps, you get a regular house ~~[illegible]~~ ~~[illegible]~~ from the first builder a small castle from second and a palace from the third.

Web crawler:
Sometime called a spider or spiderbot and often shortend to a crawler that systematically browses the world wide web and that is typically oprated by search engine for purpose of web indexing

crawler is a computer program that ~~[illegible]~~ automatically serch documents on the web. Crawlers are programmed for ~~[illegible]~~ repetitive actions so that browser is automated search engines ~~[illegible]~~ use crawlers most frequently to browse the internet and build an index

Date:..........................

- it starts with the list of urls to visit called the seeds, it identifies all the hyperlinks in the page and adds them to the list of visited urls called crawl frontier
- Urls from frontier are recursively visited according to a set of policies.

Algorithm:

```
Initilize queue (Q) with initial set of known urls
until Q empty or page or limit time ~~[illegible]~~
exhausted.
    Pop url L from front of Q.
    If L is not html page (.gif, .pdf) exit loop
    If already visited L, continue loop
        (get next url)
    Download page, P, for L
    If can't download P (error, robot exclud
        exit loop, else.

Index P (e.g add to inverted index or
    store cached copy)
Parse P to obtain list of new links N
Append N to the end of Q.
```