

Development

Intro to software engineer



Agenda



**Development
phase**



Use the right tool



**Selecting
algorithm**



Top down design



Commenting code

What is Development?

coding phase that use
Design document and
SRS as a material



Use the right tools(1)



Programming Language

Provide a set of instruction



Hardware

Provide fast computer with lot of memory and space



Network

Free access to the Internet if possible



Your Logo or Name Here



State of Development Ecosystem Infographic



Use the right tools(2)

Development environment



IDE
Debugger
Interpreter

Source code control



Tracking on changing of code

Testing tool



Make testing a unit to whole system

Use the right tools(3)



Source code formatters

Format code to make it easier to understand



Refactoring tools

Rearranging code to make it easier to read



Training

Courses, books in personal training

Selecting algorithm



Effective

- Can solve your problems
- If you write algorithm by your own perform extra test



Efficient

- Must satisfy
 - Speed
 - Memory
 - Disk space
 - requirement

Characteristics of good algorithms



Predictable

Can predict the result
when you saw the input



Simple

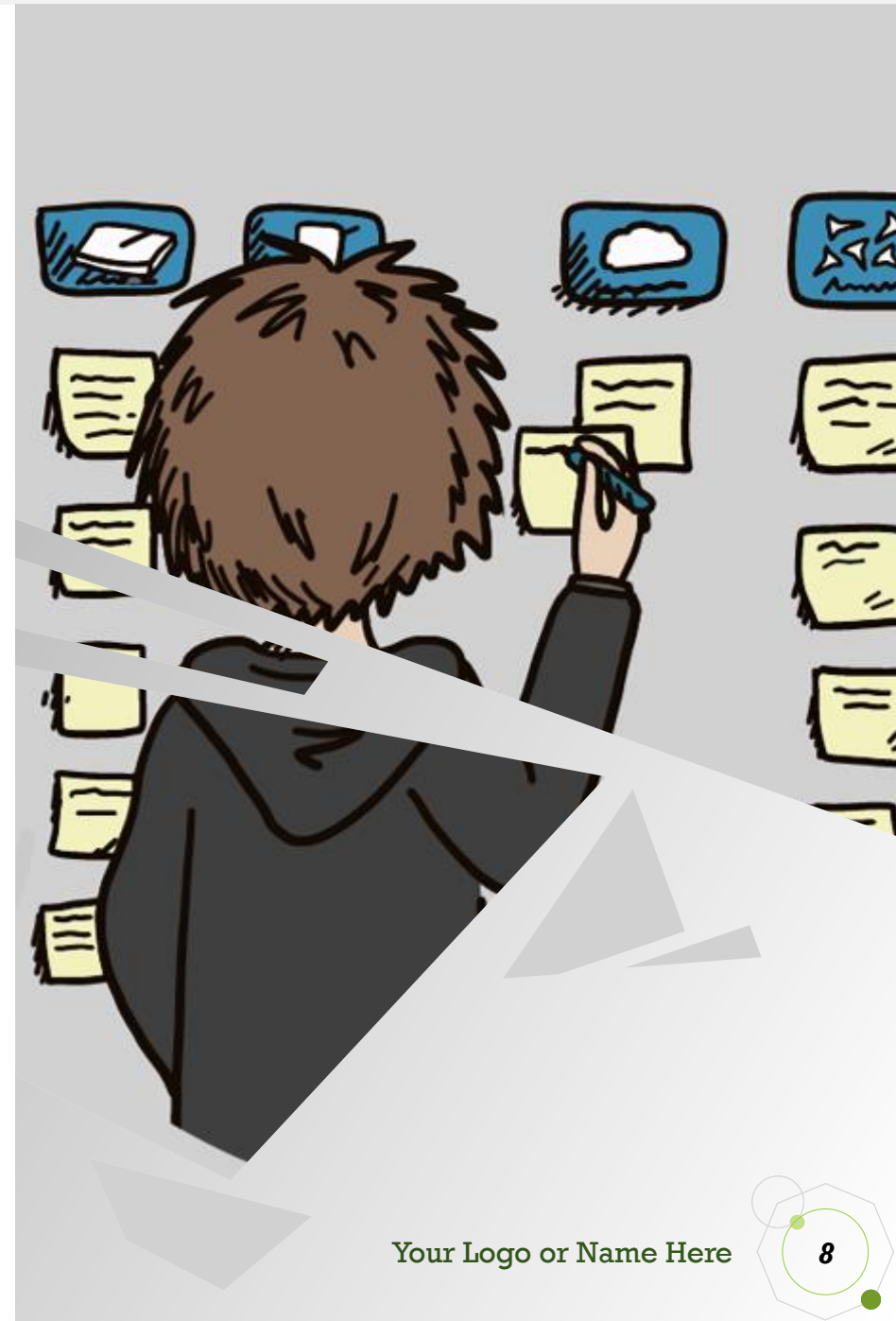
Easy to understand or
easy when explain



Prepackaged

If you can find an
algorithm that
implemented just use it

“There no need to write, test, debug, and maintain your
own code if someone else can do it for you”

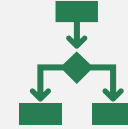


Your Logo or Name Here



Top down design

algorithm



If you write your own algorithm
consider stepwise refinement

High-level breakdown



Start with a high-level and break
down to detailed pieces

Examine the pieces



Continue examine the pieces then
list of step you need to perform to
solve the original problem

Example

- Let say we need to write the program for PromoteSales() so we plan as this

PromoteSale() (We plan the step and though how customer interact with our function)

1. For each customer:

A. If the customer is likely to buy:  IsCustomerLikelyToBuy()

i. Send e-mail, flyer, or text message depending on the customer's preferences

 SendSaleInfo()

PromoteSale() (Refine the code)

1. For each customer:

A. If IsCustomerIsLikelyToBuy()

i. SendSaleInfo()



12

Example

(We continue think many way as possible and then refine the code)

`IsCustomerLikelyToBuy()`

1. If (customer earns more than \$50,000) return `true`.
2. If (customer lives within 1 mile of a golf course) return `true`.
3. If (customer is a country club member) return `true`.
4. If (customer wears plaid shorts and sandals with spikes) return `true`.
- ...
73. If (none of the earlier was satisfied) return `false`.

Assumptions

(After we finish the break down then we do the code)

Programming tips and tricks

Tip

- Be alert of new technology
- Write for people not for the computer
 - Use meaningful names for variables
 - Indent your code nicely
 - Use comment or spell words correctly

Trick

- Comment first
- Self-document code

Where to comment



Top of any program file

- Header comment
- Include all information
 - Who wrote it
 - Why wrote it
 - When it used
 - What it should do

Above every function

- Function header
- Provides information
 - Purpose of this function
 - Parameter require
 - Value to return

In line

- Write where code is not self documenting
- If it cannot be on the same line put the comment somewhere and then explain the code

Header comment

```
/**
 * File:      compute_blackjack_odds.C
 *
 * Author1:    H. James de St. Germain (germain@eng.utah.edu)
 * Author2:    Dav de St. Germain (dav@cs.utah.edu)
 * Date:       Spring 2007
 * Partner:    I worked alone
 * Course:     Computer Science 1000
 *
 * Summary of File:
 *
 * This file contains code which simulates a blackjack game.
 * Functions allow the user of the software to play against the
 * "casino", or to simulate the odds of successfully "hitting"
 * given any two cards.
 */
```

Function comment

```
/**
 *
 * void sort( int array[] )
 *
 * Summary of the Sort function:
 *
 *     The Sort function, rearranges the given array of
 *     integers from highest to lowest
 *
 * Parameters    : array: containing integers
 *
 * Return Value : Nothing -- Note: Modifies the array "in place".
 *
 * Description:
 *
 *     This function utilizes the standard bubble sort algorithm...
 *     Note, the array is modified in place.
 */

void
sort( int array[] )
{
    // code
}
```

In line comment

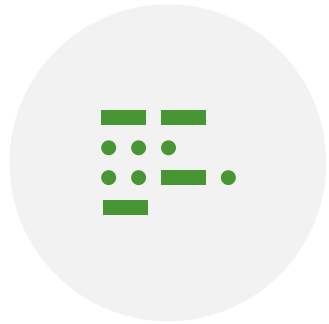
```
// Randomize the array. ← Tell the code's goal
// For each spot in the array, pick a random item and swap it into that spot.
for (int i = 0; i < items.Length - 1; i++) ← Tell how the code does it
{
    int j = rand.Next(i, items.Length);
    int temp = items[i];
    items[i] = items[j];
    items[j] = temp;
}
```

Validate Results



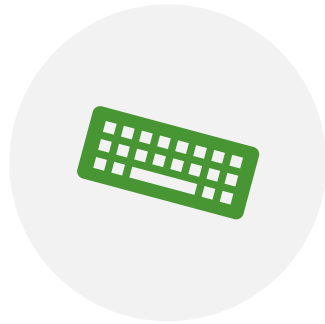
Murphy's law

Anything can go wrong
will go wrong



Validation code

Should look for trouble all
over the place



Examine the input

Make sure that input are
correct
(format, value, order)



Verify the result

Make sure that your
output is right on
testcases



Correct the calculation

Make sure that you use
the right algorithm to
manage the input and get
the output

Main tool for validating code is the assertion

Statement

- The program and its data that is support to be true

```
int n = Integer.parseInt(in.readLine());  
// what ever the input type are it will force  
// to be an Integer
```

Exception

- Throws an exception to tell you that something is wrong

```
try{  
    int n = Integer.parseInt(in.readLine());  
    System.out.println(n);  
}catch(NumberFormatException e){  
    System.out.println(e.getMessage());  
}
```




The End

This Photo by Unknown Author is licensed under CC BY

Summary

Summary tagline or sub-headline

- In summary Development phase is coding phase that using Design document and SRS as a material
- You must pick the right tool or consider the tool before you do code
- Select the right algorithm that suit to your requirement
- Validate result before send the code to tester

