WSA | Forward looking IT finishing school

Problem Solving

JavaScript

www.webstackacademy.com

Problem Solving

Problem Solving – What?

- An approach which could be taken to reach to a solution
- The approach could be ad hoc or generic with a proper order
- Sometimes it requires a creative and out of the box thinking to reach to perfect solution

Problem Solving

- Introduction to SDLC
- Polya’s Rules
- Algorithm Design methods

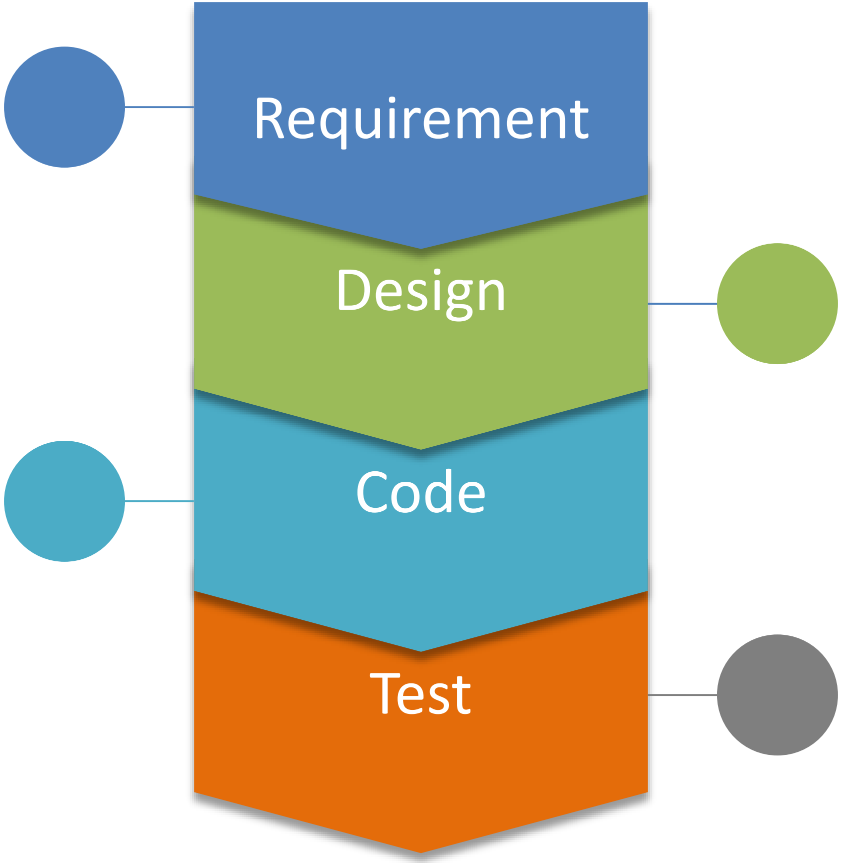
Problem Solving

- A Quick introduction to SDLC

- Never jump into implementation – Why?

  - You might not have the clarity of the application
  - You might have some loose ends in the requirements
  - Complete picture of the application could be missing and many more...

SDLC - Phases



Requirement

- Understand requirements properly
- Consider all possible I/P and O/P
- Know boundary conditions
- Get it verified

Design

- Have a proper design plan
- Use some algorithm
- Use pen and paper method
- Use flow chart (if required)
- Make sure all cases are covered

Code

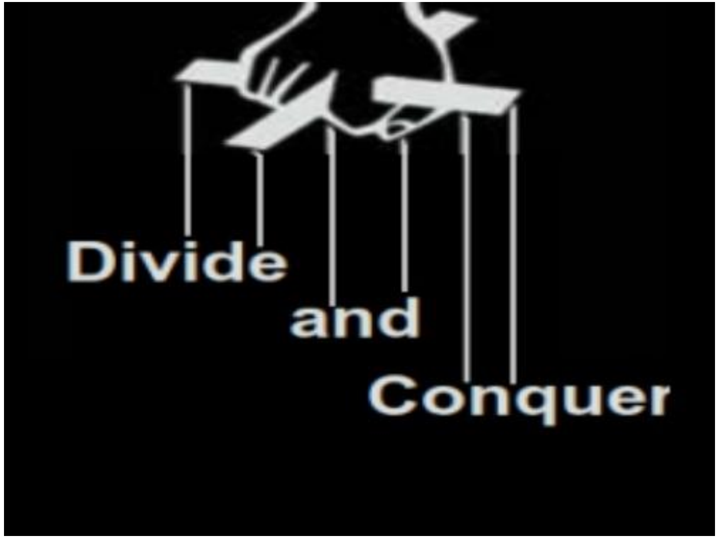
- Implement code based on the design
- Try to have maximum modularity
- Coding practices like – Indentation, Commenting, Variable naming convention, Neat file and function headers

Test

- Test the implementation thoroughly
- Capture all possible cases
- Have neat output presentation
- Requirements -> Output trace

Problem Solving – How?

- Polya's rule:
  - Understand the problem
  - Devise a plan
  - Carry out the plan
  - Look back



Problem Solving – Algorithm – What?

- A procedure or formula for solving a problem

- A sequence of unambiguous instructions for solving a problem. This means obtaining a required output for any legitimate input in a finite amount of time

- Algorithms is needed to generate correct output in finite time in a given constrained environment:
  - Correctness of output
  - Finite time
  - Better Prediction

- Steps to take:
  - Natural language (Step-by-step)
  - Pseudo codes
  - Flowcharts

Problem Solving – Daily life example

- Let's consider a problem of reaching this room

- The different possible approach could be thought of
  - Take a Walk
  - Take a Bus
  - Take a Car
  - Let's Pool

- Lets discuss the above approaches in bit detail

Algorithm – Reaching this room – Take a walk

- The steps could be like
  - Start a 8 AM
  - Walk through street X for 500 Mts
  - Take a left on main road and walk for 2 KM
  - Take a left again and walk 200 Mts to reach
- Pros
  - You might say walking is a good exercise :)
  - Might have good time prediction
  - Save some penny
- Cons
  - Depends on where you stay (you would choose if you stay closer)
  - Should start early
  - Would get tired
  - Freshness would have gone

Algorithm – Reaching this room – Take a bus

- The steps could be like
  - Start a 8.30 AM
  - Walk through street X for 500 Mts
  - Take a left on main road and walk for 100 Mts to bus stop
  - Take Bus No 111 and get down at stop X and walk for 100 Mts
  - Take a left again and walk 200 Mts to reach
- Pros
  - You might save some time
  - Less tiredness comparatively
- Cons
  - Have to walk to the bus stop
  - Have to wait for the right bus (No prediction of time)
  - Might not be comfortable on rush hours

Algorithm – Reaching the room – Take a car

- The steps could be like
  - Start a 9 AM
  - Drive through street X for 500 Mts
  - Take a left on main road and drive 2 KM
  - Take a left again and drive 200 Mts to reach+

- Pros
  - Proper control of time and most comfortable
  - Less tiresome

- Cons
  - Could have issues on traffic congestions
  - Will be costly

Algorithm – Reaching the room – Let’s pool

- The steps could be like
  - Start a 8.45 AM
  - Walk through street X for 500 Mts
  - Reach the main road wait for you partner
  - Drive for 2 KM on the main road
  - Take a left again and drive 200 Mts to reach
- Pros
  - You might save some time
  - Less costly comparatively
- Cons
  - Have to wait for partner to reach
  - Could have issues on traffic congestions

Algorithm – Reaching the room – Conclusion

- All the above solution eventually will lead you to this room
- Every approach some pros and cons
- It would be our duty as a designer to take the best approach for the given problem

Algorithm – A computer example

- Let's consider a problem of adding two numbers

- The steps involved :

  - Start
  - Read the value of A and B
  - Add A and B and store in SUM
  - Display SUM
  - Stop

- The above 5 steps would eventually will give us the expected result

Algorithm – Pseudo code

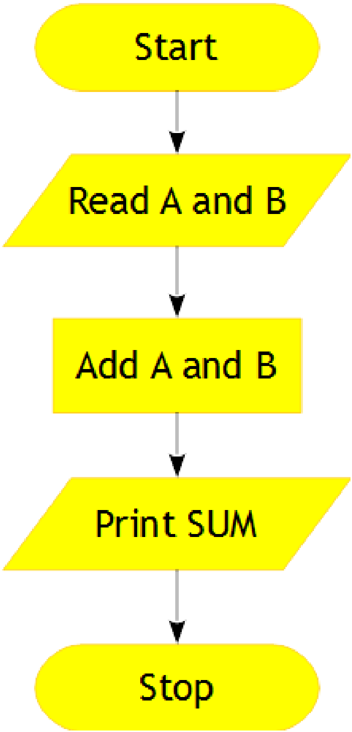
- Let's consider a problem of adding two numbers

- The steps involved :

```
BEGIN
Read A, B
SUM = A + B
Print SUM
END
```

- The above 5 steps would eventually will give us the expected result

Algorithm – Flowcharts



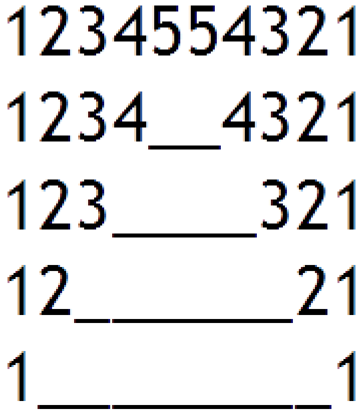
Algorithm – DIY - Pattern

- Write an algorithm to print the below pattern

```
***********
*    *
*    *
*    *
*    *
*    *
*    *
***********
```

Algorithm – DIY - Pattern

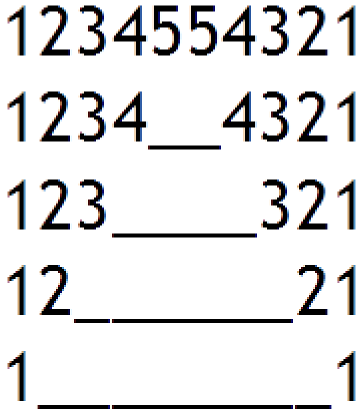
- Write an algorithm to print the below number pyramid



```
1234554321
1234__4321
123____321
12______21
1________1
```

Algorithm – DIY - Pattern

- Write an algorithm to print the below number pyramid



```
1234554321
1234__4321
123____321
12______21
1________1
```

Algorithm – DIY – Try the following

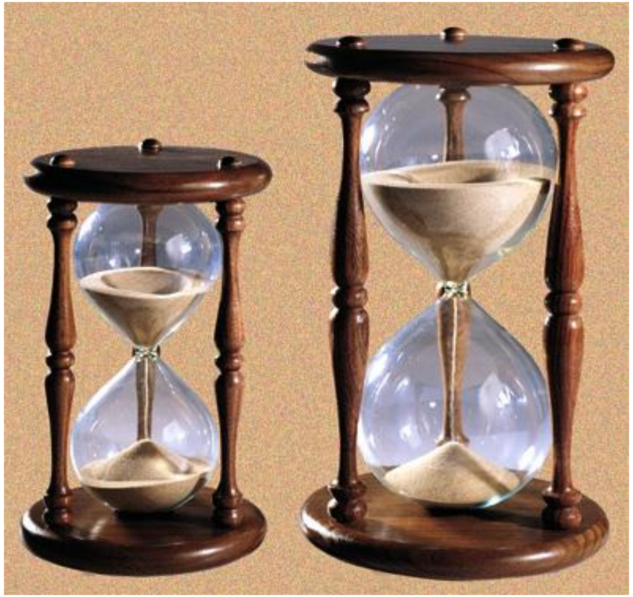
- Finding largest of 2 numbers
- Find the largest member of an array
- Count the number of vowels
- Count the number of occurrences of each vowel
- To find the sum of n – natural numbers
- Convert a number from base 10 to base N

Algorithm – DIY – Try the following

- Finding largest of 2 numbers
- Find the largest member of an array
- Count the number of vowels
- Count the number of occurrences of each vowel
- To find the sum of n – natural numbers
- Convert a number from base 10 to base N

Algorithm – DIY – Try the following

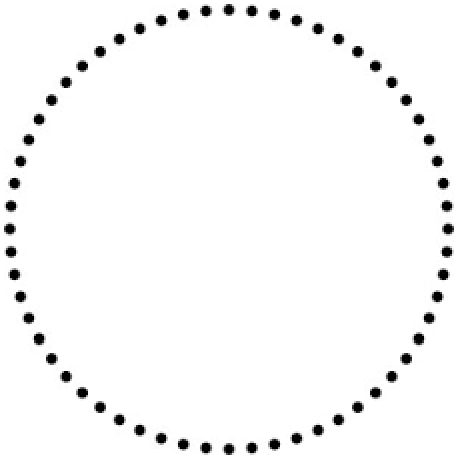
Sand Timers: You have two sand timers, which can show 4 minutes and 7 minutes respectively. Use both the sand timers (at a time or one after other or any other combination) and measure a time of 9 minutes.



Algorithm – DIY – Try the following

Mad men in a circle: N persons (given as the input by the user) are standing in a circle. They are labeled from 1 to N in clockwise order. The first person is holding a gun, will shoot a person on his right and pass the gun to the next person who is alive.

- Assuming N = 100, what is the index of the last person who will be alive?
- Can you write a generic algorithm which can handle N value up-to 1000?
- Can you validate your output by writing a JavaScript program?



WSA | Forward looking IT finishing school

Thank you

WebStack Academy

#83, Farah Towers,
1st Floor, MG Road,
Bangalore – 560001

M: +91-809 555 7332
E: training@webstackacademy.com

WSA in Social Media:

in slideshare