

9.18

(Punnet 609)

Logic in programming and thinking

- Boolean logic :
mathematical logic discovered by George Boole

OR -- any data point present

AND -- all data points are present

NOT -- one data point is present but not other

XOR -- one true and only one true

- Logic in program :

1. Procedures : sequences of instruction
2. Instructions: statements in a computer order the computer to do one thing
3. Decisions :logical action that choose one direction to go

- More logic in programs

1.loops: sequence of actions that get repeated until a conditional tells the programs that it's time to stop repeating.

2.Iteration : repeated action in a loop

3.algorithm : a set procedure which accomplish a goal

- Thinking Logically :

Always Inference

9.19

(Punnet 609)

Planning, Concurrency, and Abstraction

- Planning :

1. When planning a process, it is important to think about what data or conditions are required at certain points in the program.
2. Prefetching means getting a data that is needed before it is needed

Gantt charts:

1. Scheduling diagram that shows what tasks need to be completed in a project and when.
2. It also shows what tasks depend on other tasks and can be used to estimate how long a project will take.
3. The Critical Path is the set of tasks that depend on each other and take the longest time from beginning to the end. This is often determines how long a project takes.

- Pre and post condition:

1.Preconditions are things that must be true before the process

2.Post condition are things that must always be true just after the execution of some section of code or after an operation in a formal specification.

- Exceptions are errors cause by the application

- Concurrent processes (conjuring at the same time)

- Abstraction: hide all the details that users don't need

HWK: Why abstraction is important for the abstraction?

--- Because in that way people who don't have prior knowledge are able to make use of the program.

9.25

Bobble search:

```
int_array = [13,26,36,47,33,876,44,3,7,8,24,76,543]
index = 0
switched = True
while (switched == True):
    switched = False
    for index in range (0,len(int_array)-1):
        while int_array[index] > int_array[index + 1]:
            a = int_array[index]
            b = int_array[index + 1]
            int_array[index] = b
            int_array[index + 1] = a
            switched = True

if switched == False:
    print(int_array)
```