IIT Madras

ONLINE DEGREE

Pseudocode: Using Procedures

Analysis of top students

- Is there a single student who is the best performer across subjects?

Analysis of top students

- Is there a single student who is the best performer across subjects?
- Is the highest overall total the same as the sum of the highest marks in each subject?

Analysis of top students

- Is there a single student who is the best performer across subjects?
- Is the highest overall total the same as the sum of the highest marks in each subject?
- Need to compute maximum for different fields in a score card
    - Maths, Physics, Chemistry, Total

Analysis of top students

- Is there a single student who is the best performer across subjects?
- Is the highest overall total the same as the sum of the highest marks in each subject?
- Need to compute maximum for different fields in a score card
    - Maths, Physics, Chemistry, Total
- Ideally suited to using procedures
    - Same computation with a parameter to indicate the field of interest

Finding the maximum in a given field

- As usual, keep track of the maximum using a variable
  - Initialize to 0
  - Update whenever you see a bigger value
- The value to be compared is not fixed
  - Parameter fld determines the field of interest

```
Procedure Maxmarks(fld)
    Maxval = 0
    while (Pile 1 has more cards) {
      Pick a card X from Pile 1
      Move X to Pile 2
      if (X.fld > Maxval) {
        Maxval = X.fld
      }
    }
    return(Maxval)
end Maxmarks
```

Analysis of top students

- Use the procedure Maxmarks to find maximum marks in different categories
    - Four procedure calls, with fld set appropriately
    - Save each return value separately
- Use saved return values to compare the maximum overall total with the sum of the maximum subject totals

```
MaxMaths = Maxmarks(Maths)
MaxPhysics = Maxmarks(Physics)
MaxChemistry = Maxmarks(Chemistry)
MaxTotal = Maxmarks(Total)
SubjTotal = MaxMaths + MaxPhysics
            + MaxChemistry
if (MaxTotal == SubjTotal) {
  SingleTopper = True
}
else {
  SingleTopper = False
}
```

Summary

- Use a procedure when the same computation is used for different situations

Summary

- Use a procedure when the same computation is used for different situations
    - Parameters fix the context

Summary

- Use a procedure when the same computation is used for different situations
    - Parameters fix the context
- Use variables to save values returned by procedures

Summary

- Use a procedure when the same computation is used for different situations
    - Parameters fix the context
- Use variables to save values returned by procedures
    - Keep track of the outcomes of multiple procedure calls

Summary

- Use a procedure when the same computation is used for different situations
    - Parameters fix the context
- Use variables to save values returned by procedures
    - Keep track of the outcomes of multiple procedure calls
- Procedures help to modularize pseudocode

Summary

- Use a procedure when the same computation is used for different situations
    - Parameters fix the context
- Use variables to save values returned by procedures
    - Keep track of the outcomes of multiple procedure calls
- Procedures help to modularize pseudocode
    - Avoid descibing the same process repeatedly

Summary

- Use a procedure when the same computation is used for different situations
    - Parameters fix the context
- Use variables to save values returned by procedures
    - Keep track of the outcomes of multiple procedure calls
- Procedures help to modularize pseudocode
    - Avoid descibing the same process repeatedly
    - If we improve the code in a procedure, benefit automatically applies to all procedure calls