Practical No. 2

Subject: Java Programming

Title: Simple program based on operators, decision making, loop statements and type casting.

1. Input two integer values from keyboard and swap their contents only when first number is greater, without using any other temporary variable.
2. Read the price of an item (rupees) in float and display number of paise.
3. Convert the Celsius temperature input from keyboard into its Fahrenheit equivalent using following formula: C = (F – 32) /1.8
4. Accept the value of money in rupees and convert it into dollars.
5. Determine the sum of following series:
   1 + 1/2 + 1/3 + …….. 1/n. take n as input
6. Print the Fibonacci series on the screen by taking n as input from keyboard:
   1 1 2 3 5 8 ……. n
7. Find squares of all the numbers from 1 to n.
8. Accept a number as input and find its sum of digits.
9. Input a number and print it in reverse order.
10. Input a long number from keyboard and count total number of even digits in it.
11. Find the number of and sum of all the integers greater than 100 and less than 200 that are divisible by 7.
12. Accept marks of 10 different students from keyboard and analyze it. That is, count total first, second and pass class students.
13. Making the use of continue statement find whether the entered number is prime or not?
14. Count how many perfect square numbers are also perfect cube numbers from 1 to 1000.
15. Determine whether the entered number is palindrome or not? (A palindrome number is read same from both front and back sides such as 56365, 812218, 121 etc.)
16. Print following output using for loop:

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

17. Count and display how many perfect square numbers are palindrome numbers from 1 to 100000.
18. Input the clock time and find the angle made between the ticks of the analog clock.
19. Accept a number from keyboard and find its binary equivalent.
20. Find all the natural Armstrong numbers For example: 153 is an Armstrong that is,
    $1^3 + 5^3 + 3^3 = 153$.

Prof. B.B. Jagadale
(Subject Teacher)

Practical No. 3

Subject: Java Programming (9113) **Class: TE-CSE**

Title: Define a class and instantiate its object.

1. Define a class '**BankAccount'** having data members: acc_no, name and balance. Define appropriate methods to initialize and display the values of these data members. Input values for two objects and determine who is having more balance in his/her account.
2. Define a class '**Monitor'** having data members: company, type and resolution. Define appropriate methods to initialize and display the values of these data members. Input values for two objects and determine which monitor is better by comparing their maximum resolution.
3. Define a class '**Tennis'** having data members: player_name, country_name and total_championships. Define appropriate methods to initialize and display the values of these data members. Input values for three objects and determine who is real champion.
4. Define a class '**Month'** having data members: name, total_days and total_holidays. Define appropriate methods to initialize and display the values of these data members. Input values for two objects and determine which month is having maximum working days.
5. Define a class '**Cone'** having data members: height and radius. Define appropriate methods to initialize and find the volume of cone. Input values for two objects and display radius of cone whose volume is greater.
6. Define a class '**City'** having data members: name, population and administration. Define appropriate methods to initialize and display the values of these data members. Input values for three objects and find average population of all three.
7. Define a class '**WicketKeeper'** having data members: name, country, total_catches and total_stumpings. Define appropriate methods to initialize and display the values of these data members. Input values for two objects and determine who has taken maximum catches per stumping.
8. Define a class '**Singer'** having data members: name, songs sung and awards. Define appropriate methods to initialize and display the values of these data members. Input values for two objects and determine who has taken maximum awards per song.
9. Define a class '**Language'** having data members: state and population. Define appropriate methods to initialize and display the values of these data members. Input values for two objects and determine which language is widely spoken.
10. Define a class '**Passport'** having data members: name, nationality and passport_number. Define appropriate methods to initialize and display the values of these data members. Input values for three objects and display it.
11. Define a class '**Nation'** having data members: name, capital and density_of_population. Define appropriate methods to initialize and display the values of these data members. Input values for two objects and find name & capital of nation who's having less density of population.
12. Define a class '**Light'** having data members: speed_in_water, speed_in_air and speed_in_vacuum. Define appropriate methods to initialize and display the values of these data members. Input values for one object and determine whether speed of light in air and in vacuum is same or not.

13. Define a class `**Person'** having data members: name, age and qualification. Define appropriate methods to initialize and display the values of these data members. Input values for two objects and determine the qualification of person who is younger.
14. Define a class `**Planet'** having data members: revolution_around_self, revolution_around_sun and total satellites. Define appropriate methods to initialize and display the values of these data members. Input values for two objects and determine which planet is nearer to the sun.
15. Define a class `**PoliticalParty'** having data members: name, total_MLAs, total_MLCs and total_MPs. Define appropriate methods to initialize and display the values of these data members. Input values for two objects and determine the total leaders of these parties are elected by citizens.
16. Define a class `**MountainPeak'** having data members: name, country and height (in meters). Define appropriate methods to initialize and display the values of these data members. Input values for two objects and display the information of peak whose height is larger.
17. Define a class `**Library'** having data members: total_books, total_mazagines and total_periodicals. Define appropriate methods to initialize and display the values of these data members. Input values for two objects and determine which library is having largest collection of all books.
18. Define a class `**Examination'** having data members: year and total_students_appeared. Define appropriate methods to initialize and display the values of these data members. Input values for three objects and determine the examination in which largest number of students appeared.
19. Define a class `**Poetry'** having data members: title, poet_name and total_verses. Define appropriate methods to initialize and display the values of these data members. Input values for two objects and determine whose poetry is longest. Display the name of poetry also.
20. Define a class `**Department'** having data members: name_of_hod, total_students and per_of_result. Define appropriate methods to initialize and display the values of these data members. Input values for two objects and determine whose average result per student is better.
21. Define a class `**Forest'** having data members: area_in_sq_cm, count_of_animals. Define appropriate methods to initialize and display the values of these data members. Input values for two objects and determine which forest is having more space per animal.

Prof. Jagadale B.B.
(Subject Teacher)

Practical No. 4

Subject: Java Programming **Class: TECSE**

Title: Constructor and Method overloading.

1. Define a class **'Department'** having data members: name_of_hod, total_students and per_of_result. Define overloaded constructors to initialize and method to display the values of these data members. Input values for two objects and determine whose average result per student is worse.
2. Define a class **'Forest'** having data members: area_in_sq_cm, count_of_animals. Define overloaded constructors to initialize and method to display the values of these data members. Input values for two objects and determine which forest is having less space per animal.
3. Define a class **'Air'** having data members: per_of_oxigen, per_of_nitrogen, and humidity. Define overloaded constructors to initialize and method to display the values of these data members. Make the use of this() to call among constructors. Input values for two objects and compare them according to value of humidity.
4. Declare a class **'Distance'** having data members dist1, dist2 and dist3. Initialize two data members using constructors and store their addition using method and display the addition.
5. Define a class **'Salary'** which will contain data members basic, TA, DA, HRA. Write a program using constructors which will initialize these values for object. Calculate total salary of the employee using the method.
6. Create a class **'Student'** which will contain variables such as roll_no, name and course. Write a program to initialize these values for the object using constructors. Accept this data for three objects and display the data using method.
7. Define a class **'Product'** having data members product_id, product_name and price. Create overloaded constructors to initialize these members. Use methods to calculate total price of two different objects.
8. Create two classes **'Test1'** and **'Test2'** which stores marks of a student. Read values of class objects and calculate average of these two tests using method.
9. Overload the method 'findWeight()' in your class to convert the weight given in ounces, pounds and tons to kilograms. Use constructor to input values.
10. Write four different definitions of the method 'calculateLength()' to convert length given in inches, feet, yards and miles to centimeters. Use constructor to initialize the object.
11. Overload the method 'convertTo()' with different parameters to convert terabytes, gigabytes and megabytes to equivalent bytes value.
12. Overload the method 'temperature()' to convert given Kelvin temperature to Fahrenheit and Celsius equivalent. Use constructor to initialize the objects.
13. Overload the method 'swap()' to swap the contents of two different class' objects.
14. Overload the method 'findVolume()' to find volume of cube, cone, cylinder and sphere. Initialize the objects using constructors. Make the use of this().
15. Create a class 'Sanitarium'. Initialize its data members name, capacity and total number of rooms using a constructor. Define method to find number of persons can be allocated for one room.

16. Define a class '**HardDisk'** having data members type, speed, interface and capacity. Overload the method 'find()' to find speed and capacity of particular hard disk.
17. Defining the PI (3.14) as static variable find the surface area of cone, cylinder and area of circle using two different overloaded methods. Use overloaded constructors to initialize the values
18. Write four different definitions of the method 'calculateLength()' to convert length given in inches, feet, yards and miles to centimeters. Use constructor to initialize the object. Write all four methods as 'static'.
19. Define a class **'College'** having data members: total students, branches and percentage of total result. Input the data using static method. Class will contain the constructor. Accept this data for two objects. Create a static method containing two object parameters and display the data of college whose result is better.
20. Create a class **'Printer'** having data members: type, speed and cost. Initialize these data members using overloaded constructors. Define a static method containing two objects as parameters to find which printer is expensive.

Prof. Jagadale B.B.

(Subject Teacher)

Practical No. 5
Subject: Java Programming Class: TECSE
Title: Wrapper classes and Command-line arguments

1. Create a menu driven program to input a number from keyboard to find its binary, octal, hex equivalent, byte value, short value, int value, long value.
2. Input any number at float values from command line and find addition of them, Divide first two numbers and check whether result is infinite or not? Compare third & fourth values are equivalent or not by converting them to float wrapper class.
3. Display max value and min of double and float wrapper classes. Input two double values in string format and convert it to double, display their hash code and check their division is whether positive infinity or negative infinity.
4. Input a single character from user and check whether it is digit, letter, lowercase letter, space char, white space or not. Also change the case of it.
5. Accept two Boolean values form command-line and perform AND, OR, XOR, and NOT operation on them and display result. If they are not passed from command-line accept them from user.
6. Accept any string from user and analyze it using character wrapper class. The analyze will check how many digits, alphabets, space chars, upper case letters and white spaces are there in the string.
7. Enter two byte values from keyboard, convert them to Byte objects, check for their equality, identify their hashcode and convert them to String format.
8. Accept five short values from command line, find average of their hash codes, check third and fourth objects are equal or not? Compare first and second using compareTo() method and convert fifth value to String equivalent.
9. Identify maximum & minimum values of Byte, Short, Integer, & Long classes. Divide max value of Double by Float and store it in double. Convert this value to double and display its byte value, float value, int value and long value.
10. Input any two float numbers from command-line and find their octal, binary, hex equivalents. Find addition of octal, binary and hex equivalents respectively.
11. Create a menu driven program to input two long numbers for: 1. Convert them to Long objects. 2. Check for their equality. 3. Find their hex equivalents. 4. Display their hash codes.
12. Create and input array of Double values. Keep 0^{th} value as NaN and 1^{st} value as and Infinity. Sort all the values a display their long values in sequence.
13. Input a String from keyboard and inverse the case of it. Count total number of digits and white space characters from it. Display max and min values for radix and type of character.
14. Input the array of Booleans and demonstrate the use of booleanValue(), equals(), hashcode(), toString(), valueOf(), parseBoolean() on it.

15. Input an array at five floats and five Integers from command-line. Find largest of them and display their sorted formats differently.
16. Input any number of Strings form command-line and check total number of special symbols and white space character in it. Also inverse the case of all.
17. Enter a number from command-line in long format and display its binary, octal, hex equivalents, byte value, short value, long value and int value. Display its hashcode and convert it back to String.
18. Accept following data of an employee using data input stream: name(string), age(byte), salary(int), male/female(string/char), annual income(long), Display name in uppercase (don't use String class' method) salary in byte value, male/female in upper case and annual income in int value.
19. Input a double value form user and demonstrate hashCode(), isInfinite(), isNaN(), parseFloat() methods on it.
20. Input a short value form command-line and demonstrate parseByte(), compareTo(), decode(), toString(), valueOf(), intValue(), shortValue() methods on it.

Prof. Jagadale B.B.

(Subject Teacher)

Practical No. 6
Subject: Java Programming Class: TECSE
Title: Program based on Strings

1. Input an array of names of ten cities. Convert them in lower case, check any two of them are equal or not and display first character of all names.
2. Accept names of your six friends as input. Append their surname with it afterwards, reverse their names and display all.
3. Input names of five players from keyboard. Display their names from index position 2 onwards; replace all the occurrences of "e" with "a" only if their names starts with letter "s".
4. Input names of seven sports from keyboard. Display those names starts with "t", display names ending with "t", insert character "m" at position 3 for all and delete character at position 4 and display all the strings.
5. Sort the names of 5 actors in descending order. Identify average length of all names, append a "*" at the end of all names.
6. Accept the names of 5 films from keyboard. Display the names starting with "k" ending with "a" display substring of all from 1^{st} to 3^{rd} position and find last occurrence of "e" for all the names.
7. Input names at six states and change their first letter with "n" if it is "m" and display them. Append string "state" to the end of all; insert string "India" from 4^{th} position for all the names.
8. Input names of five subjects from user and display "Programming" subjects, display them in lower case by trimming them, reverse the contents of all.
9. Accept names of five school subjects, sort them, delete their last character and insert character "m" at position 10. Make length at all subjects to five by truncating it.
10. Input names of five actresses form user and check if any one of them is equal or not by ignoring case. Convert the names to char array if they starts with letter "M" display character sequence. Delete all vowels from all names.
11. Input names of five TV channels and display those owned by "Star", insert "*" at the start of all those and delete word "star" from them. Display all the strings.
12. Accept names of five languages display names of languages ending with "i", delete their "i" and insert "a" instead at that, convert all names in upper case and display them.
13. Input names at five engineering branches; replace all occurrences of "C" with "c", display names starting with "C", set character at 5^{th} position with "o" and identify last occurrence of letter "i" in all names.
14. Input names of six computer service companies from user, count average length of all, display names ending with "i", display first 3 characters of all,

display the strings at even index position in upper case and remaining in lower case.

15. Enter names of five nations; categorize nations with name ending with "land" and "ia" display them. Find longest string, Find nations starting with "Ne" and display names in reverse order.
16. Input the names of seven soaps from user, sort them by ignoring the case, check whether any two of them are equal or not, display content of all by deleting the last character.
17. Enter the names of seven villages from keyboard display "gaon" and "pur" differently. Replace "gaon" with "pur" and vice versa, identify their substrings from 2^{nd} position to end of string and display all names in reverse order.
18. Input the names of five News papers, calculate total no. of spaces in all names, change character of third position with "K", delete first character of all, insert "c" instead of it and display all names in reverse order.
19. Accept names of six books from keyboard, sort names, starting with a consonant, calculate total length of all, find index of "J" from position 2 display the names.
20. Input any five arithmetical expressions from user, count total no. of operators append first two and last two expressions, insert third expression in last two and display all the expressions.
21. Create a menu based program to input two strings and perform following operations:-
    1. Find length of both.
    2. Concert he strings.
    3. Find equality.
    4. Find greater string.
    5. Print upper case concerted format.
    6. Insert first string at the center of second.

Prof. Jagadale B.B.

(Subject Teacher)

Practical No. 7
Subject: Java Programming Class: TECSE
Title: Program based on arrays.

1. Define a class **Item** containing *code* and *price*. Accept this data for five objects using array of objects and display the code of object for which having highest price.

2. Create a class to define an array of '*num*' with 10 elements. Initialize the element using constructor. Define a method *sortArray*() to sort the element of array.

3. Create a method *sortIt*(Array, boolean value). It will sort and display the array in ascending order when Boolean value is true else it will use descending order to perform this task. Define this method in different classes.

4. Initialize the array of five double values using constructor and use its method display() to display those value which are greater than 10000.5.

5. Define a class **Soccer** having elements as *player_name*, *country*, *club_name* & *no_of_goals*. Initialize data for five players and display data of those having goals more than 50.

6. Define a class **State** having data members *state_name, capital, population*. Initialize the data for five states and display name and capital of highest populated state.

7. Create a class **House** with variables as *houseName, houseNo* and *roadName*. Initialize five object and display their data in reverse order.

8. Define a class **AccountInfo** having variables as *name_of_the_bank, accountNumber, the_balance* (private). Initialize the data for five objects & calculate average of the_balance.

9. Create a class **Calculator** which contains data members as *company_name, price*, and *type*. Initialize the data members using constructor for three objects & display data for the cheapest calculator.

10. Define a class **FilmFare** with data variables as *year_of_award, cash_prize, best_film*. Initialize the data for four objects and determine which best film got highest cash prize on which year?

11. Define a class **Triangles** having data members as *type, area, perimeter*. Initialize these all private data variables for five objects and display it in tabular format.

12. Create a class **House** with instance variables as *houseName*, area_in_sqm, *no_of_doors* and *no_of_windows*. Initialize three objects with all these values and count total number of windows that contains. Display the data also.

13. Define a class **MicroProcessor** with data variables *company_name, name* and *speed*. Initialize these data for five objects and sort it in ascending with respect to speed.

14. The class **TYCO** is created with following data members: *no_of_students, no_of_girls, per_of-result*. Initialize the data for five TYCO classes and determine total no. of boys having highest percentage of result.

15. Create a class **Polytechnic** with data values as *name, noOfBranches* and *city*. Initialize theses data for five objects and display it in tabular format in reverse order.

16. Create a class **City** with values of *name*, *population* and *state*. Initialize these data for five cities and calculate total population of all.

17. Define a class **Book** having instance variables: *name, totalPages, coverPrice* and *author_name*. Initialize theses values using constructor for four objects and calculate average pages and average cover price.

18. Create a class named **Dictionary** with data: *nameofPublisher, pages* and *total_words*. Input these values using method for four objects and display these data for the dictionaries having words more than 40000.

19. Define a class **Tennis** having data variables as *player_name, maleFemale(char), totalGrandSlams*. Initialize data for five players and display the data in tabular format only for female players.

20. Create a class **MasterOfArts** with instance variables as *subject_name, passingYear* and *percentage*. Input these values using constructor for five objects and display it. Also calculate how many students passed out on year 2009?

Prof. Jagadale B.B.

(Subject Teacher)

Practical No. 8
Subject: Java Programming Class: TECSE
Title: Program based on Vector.

1. Implement the Vector as a queue. That is, create a menu driven program to add and delete the elements from the queue. Display the Vector after each action.
2. Create a menu driven program to perform following operations
    a. Insert element in Vector
    b. Delete the element at 'n'
    c. Empty the vector
    d. Check empty or not?
    e. Exit
3. Create a menu driven program to perform following operations
    a. Add elements in Vector
    b. Delete the last element
    c. Check empty or not?
    d. Display element at position 'n'
    e. Exit
4. Create a menu driven program to perform following operations
    a. Create a Vector
    b. Delete the element at 'n' position
    c. Add element at start and end
    d. Copy into array and display
    e. Exit
5. Create a menu driven program to perform following operations
    a. Create Vector of Booleans
    b. Create Vector of Strings
    c. Create Vector of Floats
    d. Merge Strings and Floats Vectors
    e. Display vectors
    f. Exit
6. Create a menu driven program to perform following operations
    a. Add element in Vector
    b. Insert the element at 'n'
    c. Find the length
    d. Empty the vector
    e. Exit
7. Create a menu driven program to perform following operations
    a. Add element in Vector
    b. Delete the element at 'n'
    c. Find first and last element
    d. Display current capacity
    e. Exit

8. Create a menu driven program to perform following operations
    a. Add element in Vector
    b. Display the size
    c. Trim to size
    d. Set size to 'n'
    e. Display the Vector
    f. Exit
9. Implement the Vector as a stack. That is, create a menu driven program to pop and push the elements of the stack. Display the Vector after each action.
10. Create a menu driven program to perform following operations
    a. Add element in Vector
    b. Copy contents into objects array
    c. Set element at 'n' position
    d. Create a clone and display
    e. Exit
11. Create a menu driven program to perform following operations
    a. Add element in Vector
    b. Display element at n^{th} element
    c. Display first and last element
    d. Remove all elements
    e. Display vector
    f. Exit
12. Create a menu driven program to perform following operations
    a. Add element in Vector
    b. Sort the vector
    c. Insert element at 'n' position
    d. Delete element at 'n' position
    e. Display current capacity and length
    f. Exit
13. Create a menu driven program to perform following operations
    a. Add element in Vector
    b. Check empty or not?
    c. Display vector
    d. Display current size
    e. Find element at 'n' position
    f. Exit
14. Create a menu driven program to perform following operations
    a. Add element in Vector
    b. Ensure capacity to 'n'
    c. Display element at 'n' position
    d. Remove all elements.
    e. Display whole vector
    f. Exit
15. Create a menu driven program to perform following operations

a. Add element in Vector
b. Find index of any element
c. Display vector
d. Display current capacity and length
e. Remove the element
f. Exit

16. Create a menu driven program to perform following operations
    a. Add element in Vector
    b. Set size to 'n'
    c. Trim to size
    d. Display current capacity
    e. Check vector is empty or not?
    f. Exit
17. Create a menu driven program to perform following operations
    a. Add element in Vector
    b. Sort the vector
    c. Copy contents into array and display
    d. Ensure minimum capacity
    e. Remove element at 'n' position
    f. Exit
18. Create a menu driven program to perform following operations
    a. Create a vector with 'n' capacity
    b. Add element in Vector
    c. Insert element at position 'n'
    d. Display first and last element
    e. Find index of an element
    f. Exit
19. Create a menu driven program to perform following operations
    a. Create vector of 'n' capacity and 'm' size
    b. Sort the vector
    c. Trim to size
    d. Ensure minimum capacity
    e. Set length to 'n'
    f. Exit
20. Create a menu driven program to perform following operations
    a. Add element in Vector
    b. Insert any Float element at 'n'
    c. Display capacity and length
    d. Display last element of the vector
    e. Remove element at 'n' position
    f. Exit

Prof. Jagadale B.B.

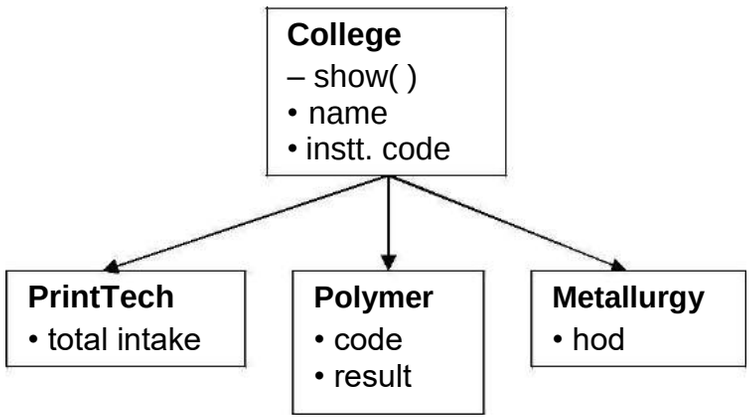
(Subject Teacher)

Practical No. 9

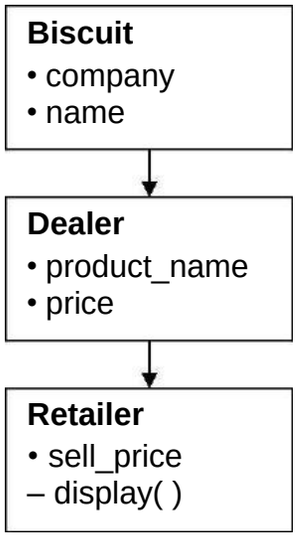
Subject: Java Programming **Class: TECSE**

Title: Single and Multilevel Inheritance.

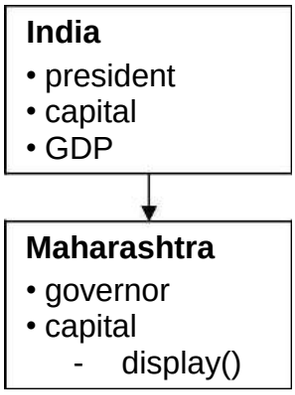
1. Declare a class **'Student'** with data members 'name', 'roll number' and 'marks'. Also declare a class **'MyRecord'** which inherited from class 'Student'. Display name, roll number and percentage by creating the object of class 'MyRecord'.
2. Create a base class to get two numbers from user in the base class and one number from derived class. The derived class contains a method 'dispMax()', which displays maximum of three numbers, a method 'dispMin()' to display minimum of three numbers and method equality( ) which checks whether any two of these three numbers are equal or not. Use constructors.
3. Implement the inheritance shown in figure below. The rectangle represents the classes. The '•' represents data members and '–' represents methods. Display information of each branch along with college information.



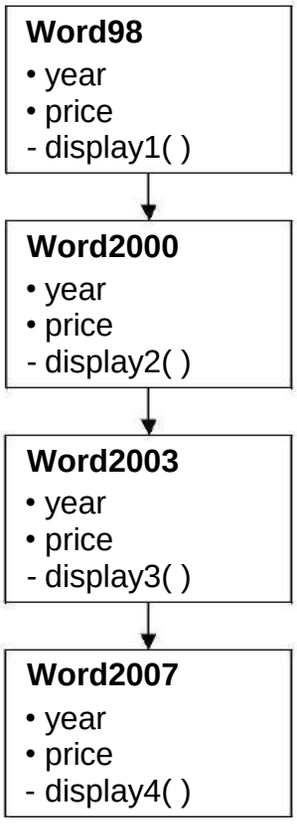
4. Create a class **'Object3D'** that contains 'dimension1' and 'dimension2' as data members. From this class, derive classes named '**Object1'** and '**Object2'** that has additional data member 'dimension3' to each. The members of the class '**Object3D'** should consists of a constructor and a method volume(). The '**Object1'** and '**Object2'** should have constructors. Write a program to find volume of two objects by creating objects of '**Object1'** and '**Object2'**. Use volume = (⅝)d1 + (d2 × d3).
5. Implement the multilevel inheritance from following figure where sell_price is price + 2.5% VAT extra. Accept and display whole data for one biscuit box.



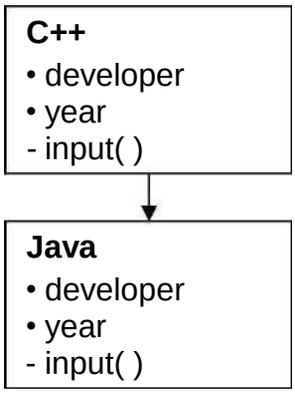
6. Create a class **'TYIF'** which contain data members: total subjects, total students. Inherit two classes '**Test1'** and '**Test2'** from it to have marks from two different MSBTE class tests. Input the values for all data members (including super class) of these classes and find average of both the class tests.
7. Implement following inheritance for one object. Input whole data using constructor of sub-class and display these data. Display capital of India only.



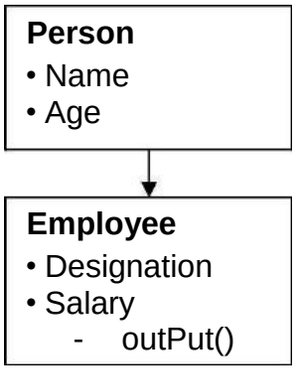
8. Implement following inheritance. By invoking the constructors of **'Word2007'** initialize the year values of all classes and by calling display4() method display the years of all classes (Hint: Make the effective use of 'this' and 'super'.



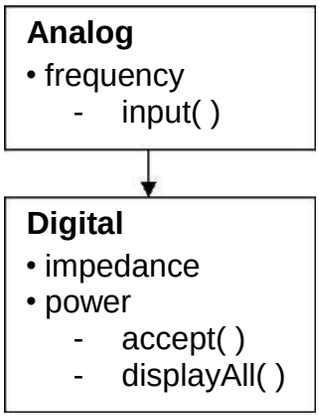
9. Implement following inheritance. Create the class of **'Java'** to initialize all values of both classes using input() and display these values by defining the display( ) in the class **'Java'**.



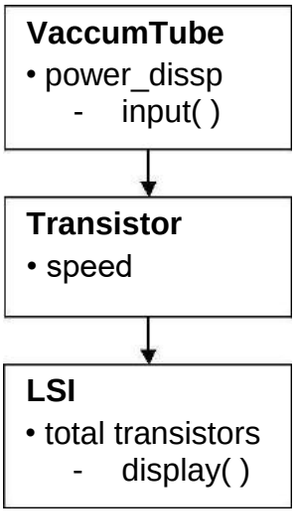
10. Define a class **'Human'** having data members name and gender. Create classes 'Woman' and 'Man' derived from '**Human'** having data member age in both. Create objects of '**Man'** and '**Woman'** having a constructor to initialize all data members (including super class) for both classes. Define method find() in 'Human' to pass objects of '**Man'** and '**Woman'**. It will find who is elder.
11. Implement following inheritance. Display data for three employees and display names of the employees having salary greater than 15000.



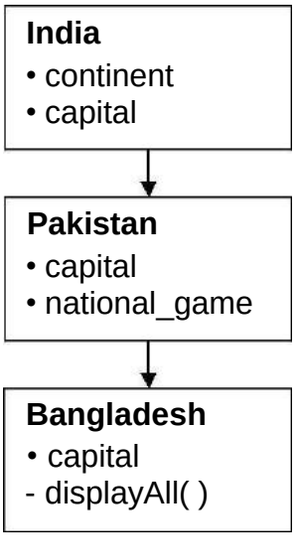
12. Implement following inheritance. Create object of '**Digital'** to use both the methods in order to initialize all data members. Assume all values are in 'double'. Display all the accepted values. Make use of 'this'.



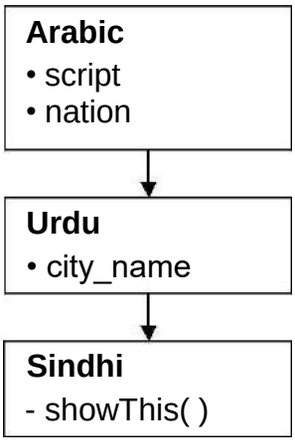
13. Implement following inheritance. Use only given methods to initialize all the data values. Make the use of super to call constructors. Display all values



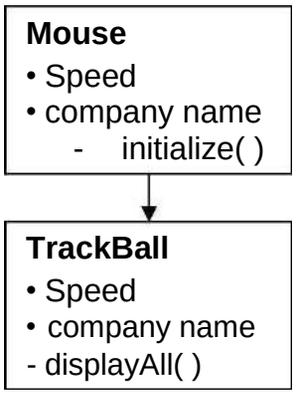
14. Implement following inheritance. Making the use of 'super' and 'this' initialize all the values by creating object of **'Bangladesh'**. Use displayAll() to display values of all data members. Assume all String values.



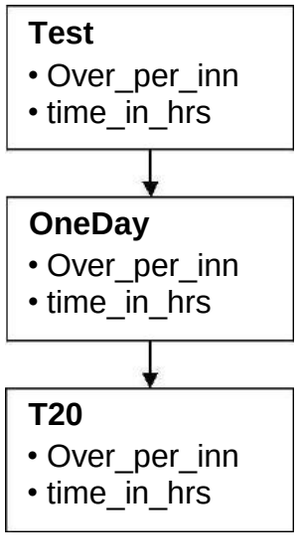
15. Implement following inheritance. Consider all members of classes as strings. Make the use of super to call constructors of classes by creating object of **'Urdu'**. Use 'showThis()' to display all initialized contents.



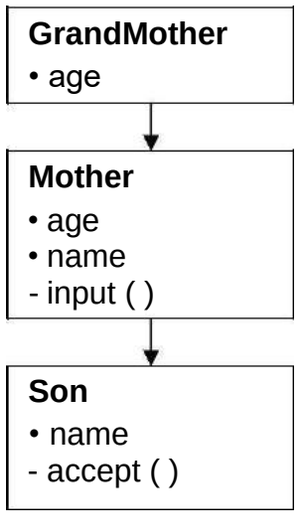
16. Implement following inheritance. Use input() to initialize **'Mouse'** object and a constructor in 'initialize( )' to initialize 'TrackBall' object. Display all these values using displayAll( ).



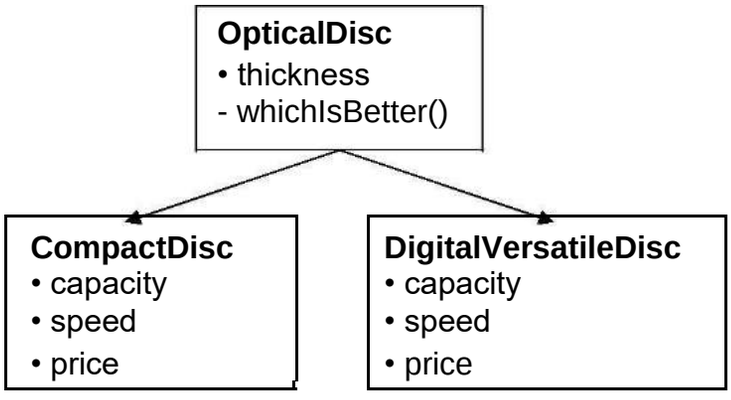
17. Implement following inheritance. Making the use of **'T20'**s object initialize all the values and define appropriate methods to display all the values.



18. Implement following inheritance. Make the use of 'super' and 'this' to initialize and call constructors. Input and output whole data for one object.

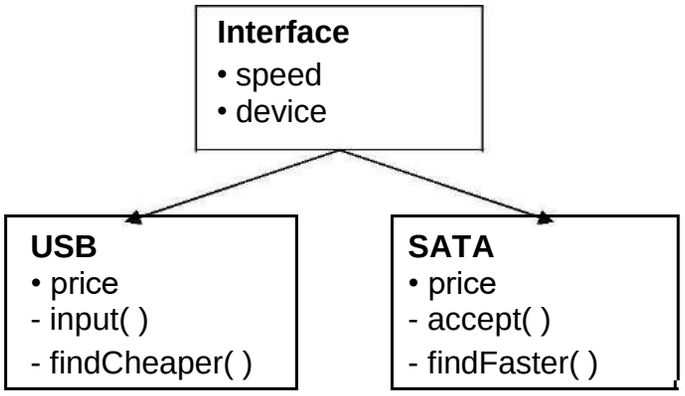


19. Implement following inheritance. Create the objects of **'CompactDisc'** and **'DigitalVersatileDisc'** to initialize all the data values. Pass the objects as parameters to method 'whichIsBetter()' to find which object is having better performance according to capacity and price ratio.

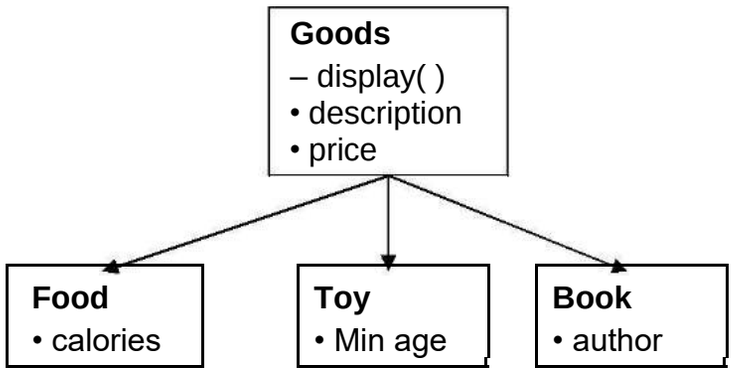


20. Implement following inheritance. Create the objects of **'USB'** and **'SATA'** to input all these values and display those using appropriate methods. The

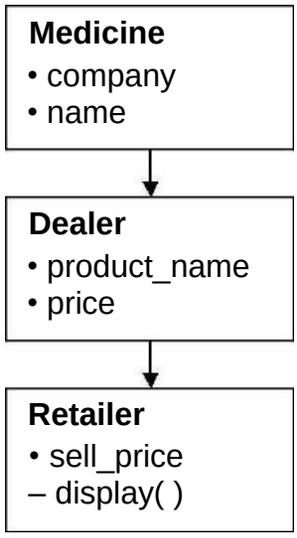
method 'findCheaper()' and 'findFaster()' accepts the objects of **'USB'** and **'SATA'** to find which interface is better by comparing in terms of price and speed respectively.



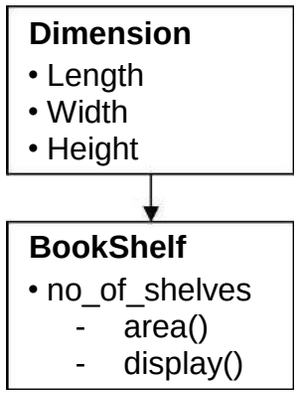
21. Declare a class 'student' with data members 'name' and 'roll number'. Also declare a class 'Record' which inherits class 'student'. Display name and roll number with address by creating the object of class 'Record'.
22. Create a base class to get two numbers from user in the base class. The derived class contains a method 'dispMax()'. Which displays maximum of two numbers and a method 'dispMin()' to display minimum of two numbers. Use constructors.
23. Implement the inheritance shown in figure below. The rectangle represents the classes. The '•' represents data members and '–' represents methods. Display information of each class.



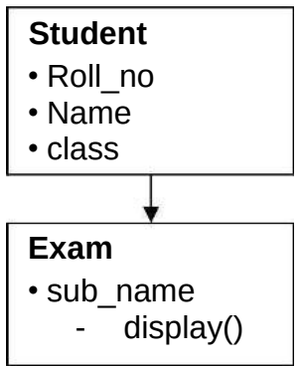
24. Create a class 'rectangle' that contains 'length' and 'width' as data members. From this class derive a class named 'box' that has additional data member 'depth'. The members of the class 'rectangle' should consists of a constructor and a method area( ). The 'box' should have a constructor. Write a program to implement this.
25. Implement the multilevel inheritance from following figure where sell_price is price + 1.5% VAT. Accept and display data for one medicine.



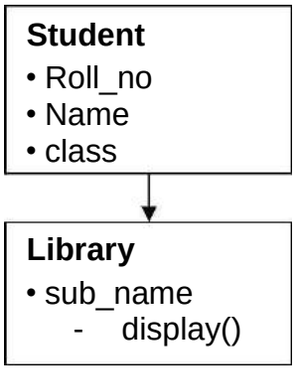
26. Display an invoice of several items. It should contain the item name, quantity, price, and total cost on each line for the quantity and item cost. Use two classes. The first class contains the item data and methods to get and set the item name, quantity and price. The other class creates objects for the items and uses the objects to call the set and get methods.
27. Implement following inheritance. Assume suitable methods.



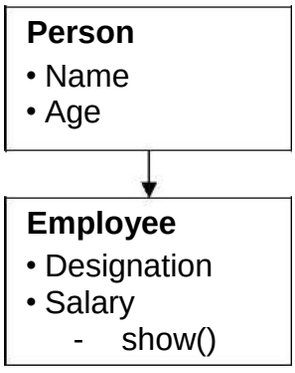
28. Implement following inheritance. Assume suitable methods.



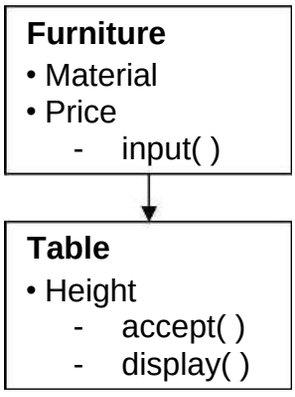
29. Implement following inheritance. Assume suitable methods.



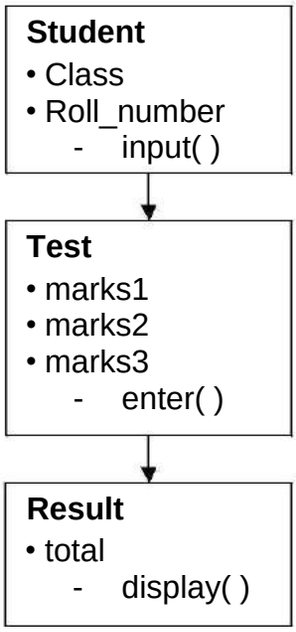
30. Define a class 'Person' having data members name, age and gender. Create class 'Student' derived from 'Person' having data members roll_number and name. Crete object of 'Person' having a constructor to initialize all data members of both classes. Display whole data using method 'show()'.
31. Implement following inheritance. Display data for three employees and display names of the employees having salary greater than 5000.



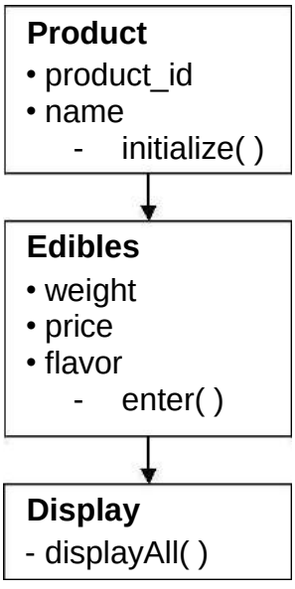
32. Implement following inheritance. Assume suitable methods.



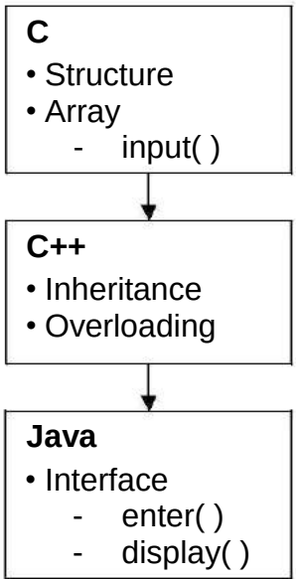
33. Implement following inheritance. Assume suitable methods. Make the use of super to call constructors.



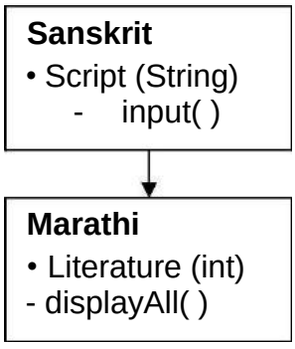
34. Implement following inheritance. Assume suitable methods. Make the use of super to call constructors.



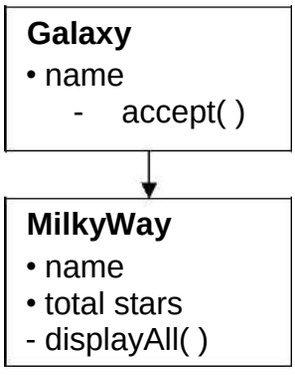
35. Implement following inheritance. Consider all members of classes as integer. Make the use of super to call constructors.



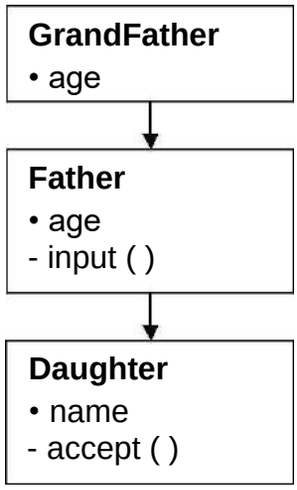
36. Implement following inheritance. Make the use of super to initialize and call constructors.



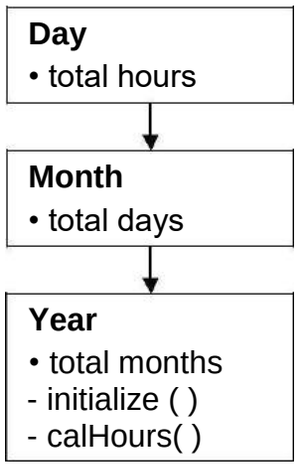
37. Implement following inheritance. Make the use of super and this to initialize and call constructors.



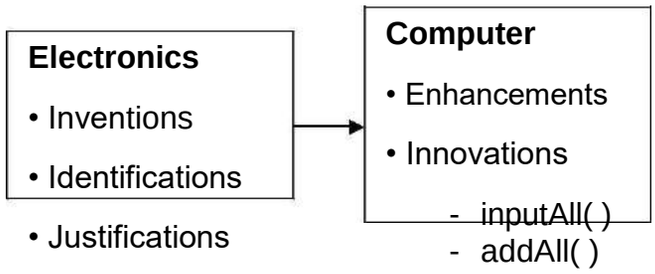
38. Implement following inheritance. Make the use of 'super' and 'this' to initialize and call constructors.



39. Implement following inheritance. Make the use of 'super' and 'this' to initialize and call constructors. Initialize one object's values and find total hours in the year.



40. Implement following inheritance. Make the use of 'super' and 'this' to initialize and call constructors. Initialize one object's values and find total hours in the year.



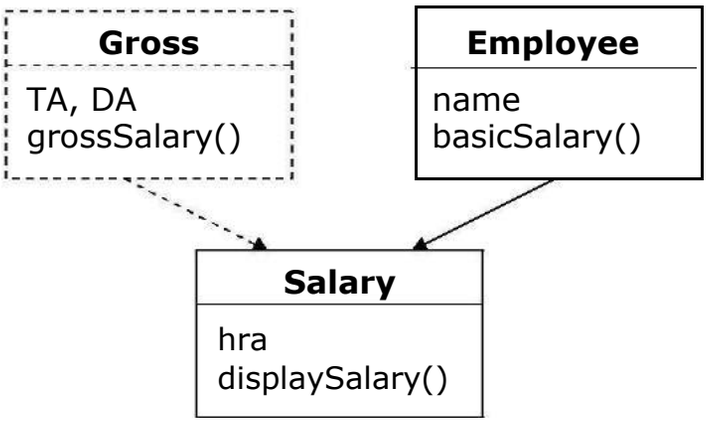
Prof. B. B. Jagadale
(Subject Teacher)

Practical No. 10

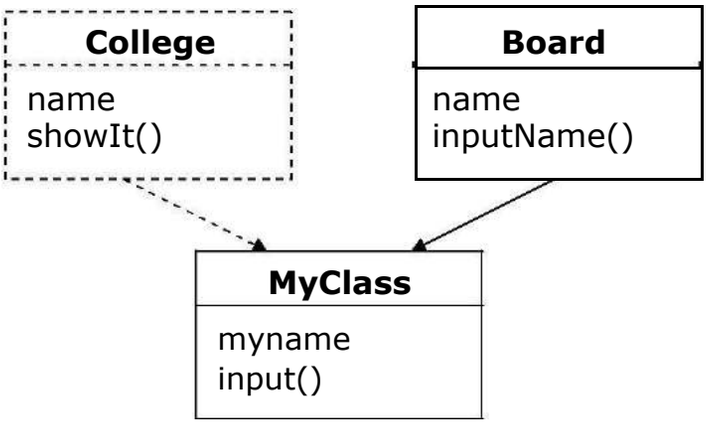
Subject: Java Programming **Class: TECSE**

Title: Multiple Inheritance using Interface.

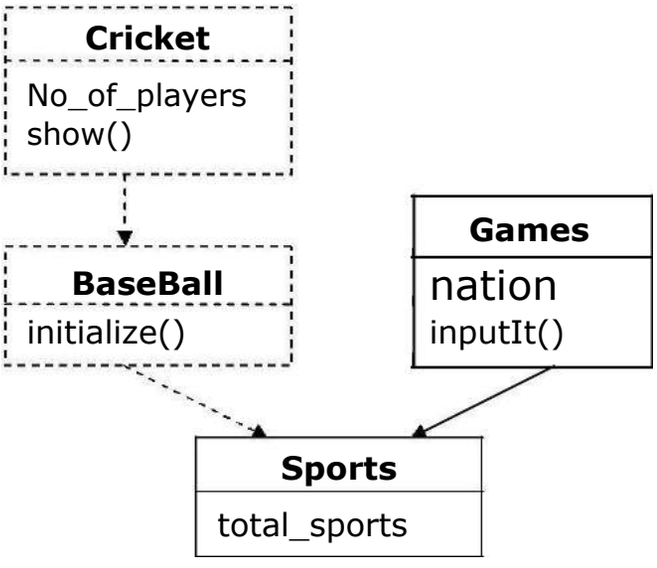
1. Implement the following inheritance:



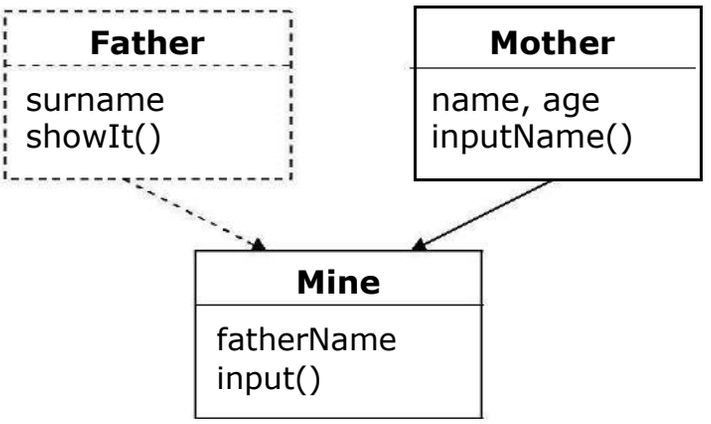
2. Implement the following inheritance:



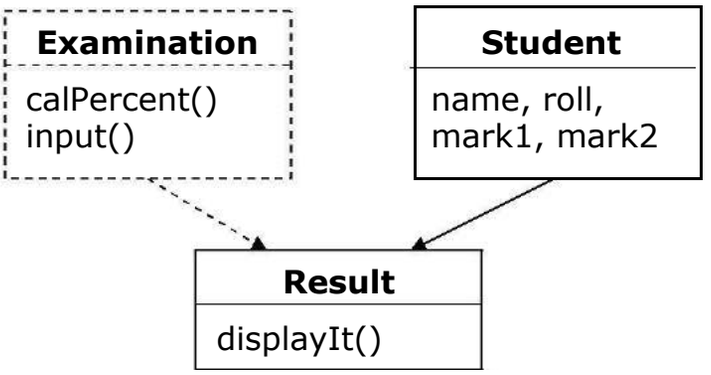
3. Implement the following inheritance:



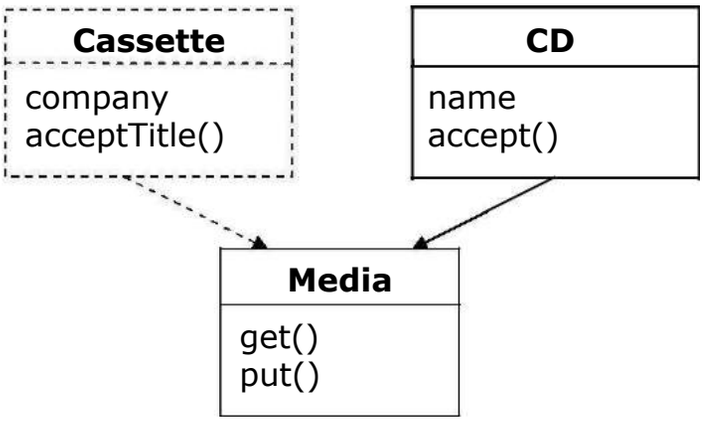
4. Implement the following inheritance:



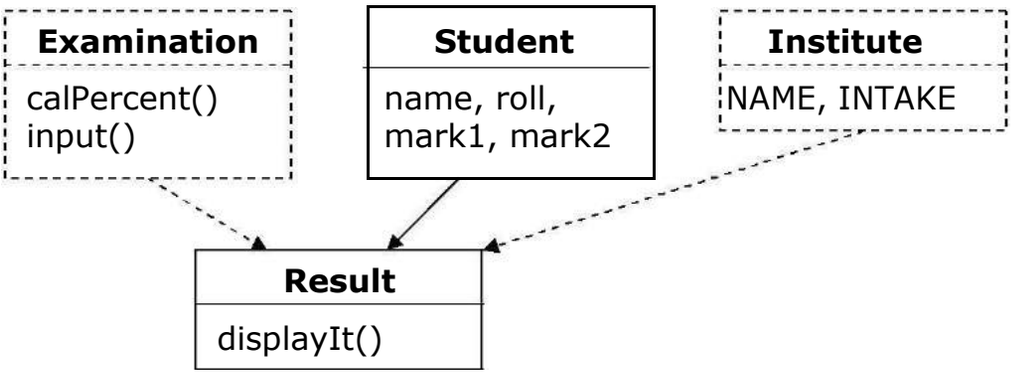
5. Implement the following inheritance:



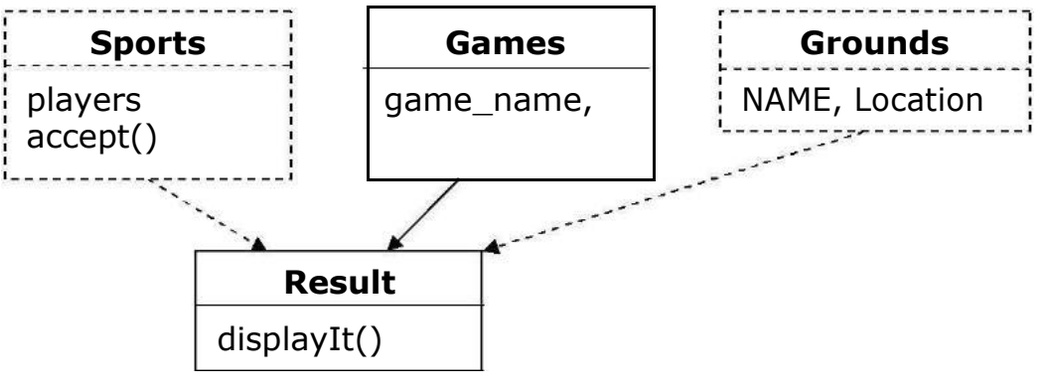
6. Implement the following inheritance:



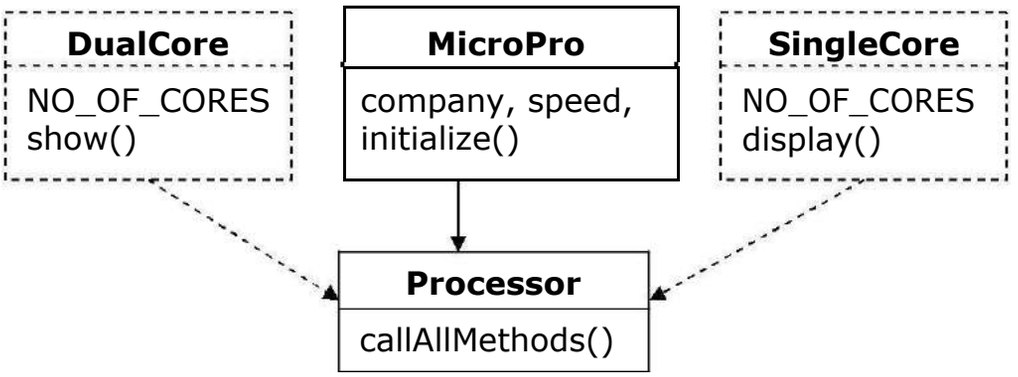
7. Implement the following inheritance:



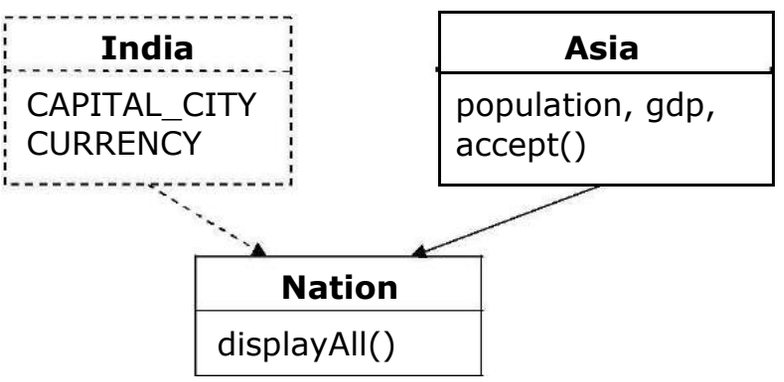
8. Implement the following inheritance:



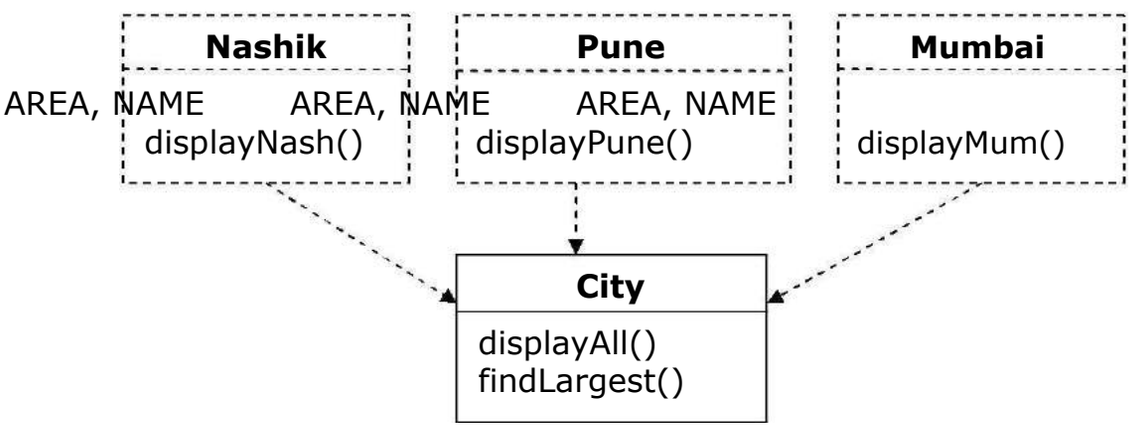
9. Implement the following inheritance:



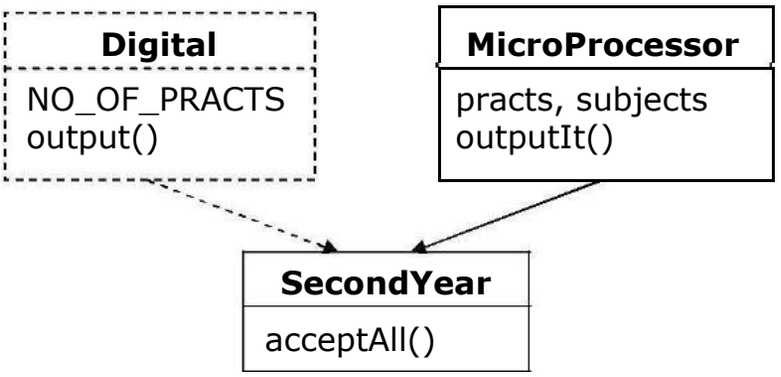
10. Implement the following inheritance:



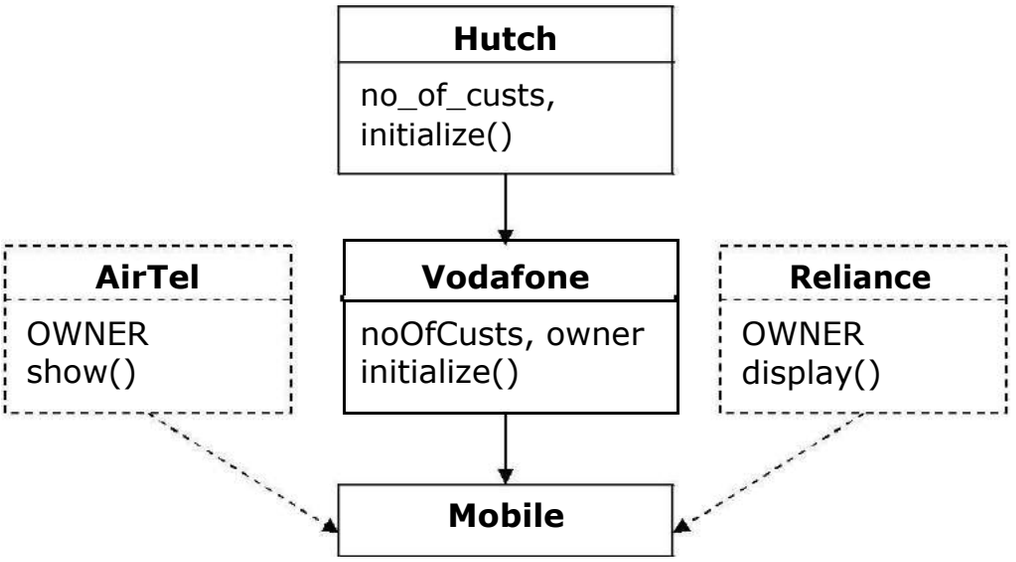
11. Implement the following inheritance:



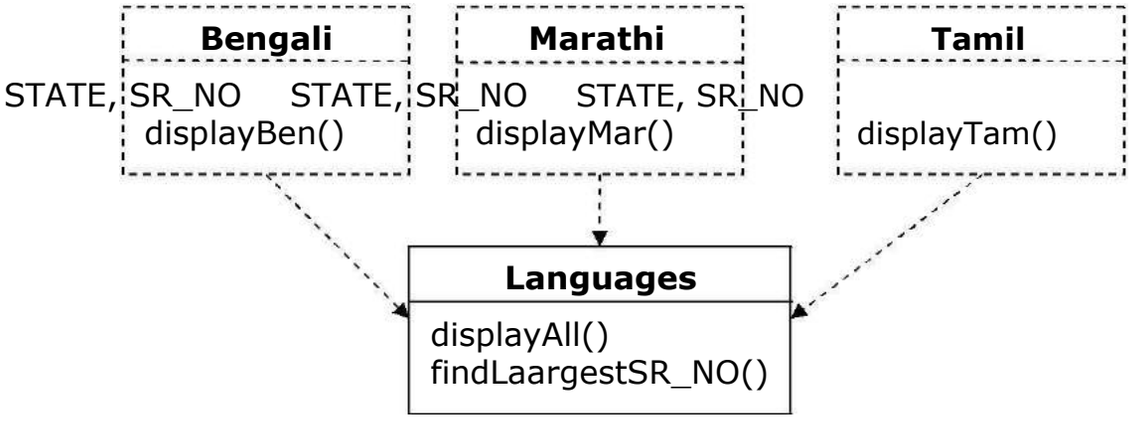
12. Implement the following inheritance:



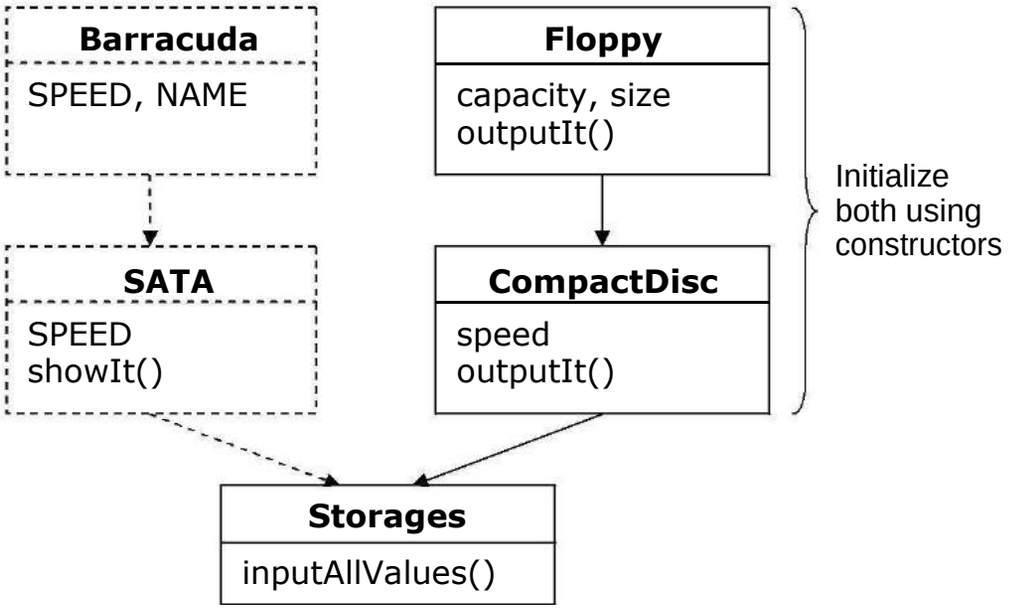
13. Implement the following inheritance:



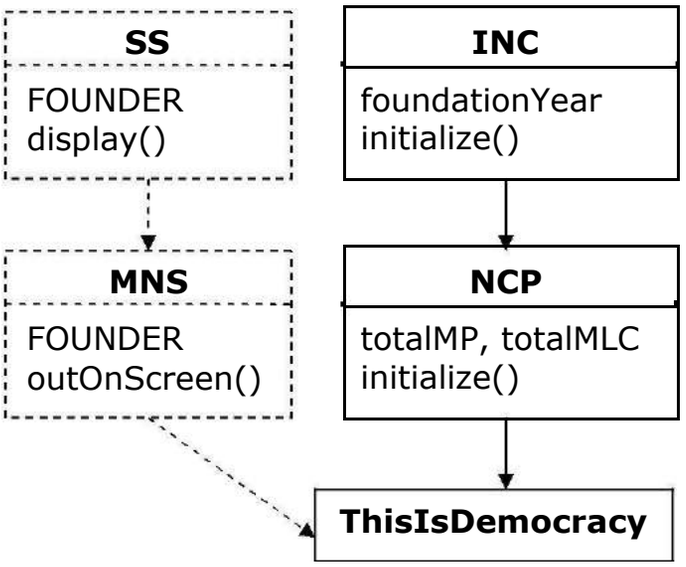
14. Implement the following inheritance:



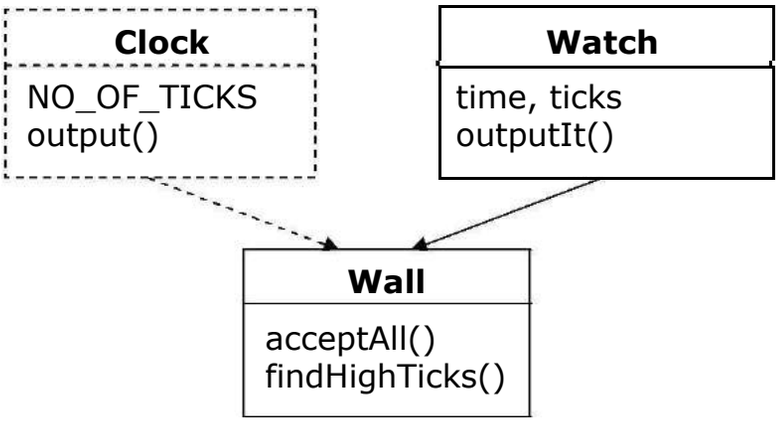
15. Implement the following inheritance:



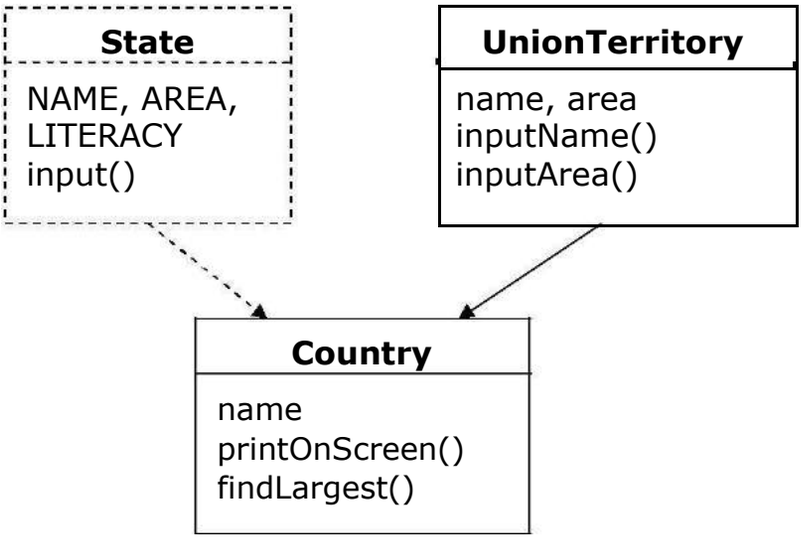
16. Implement the following inheritance:



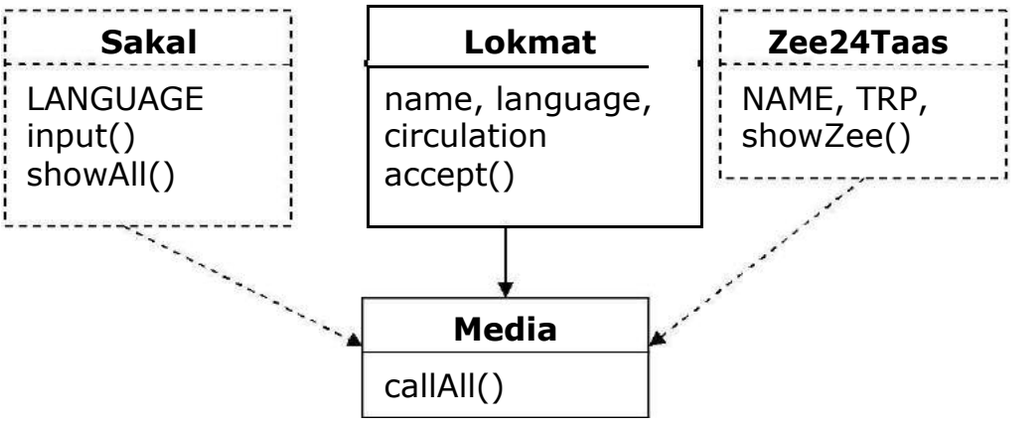
17. Implement the following inheritance:



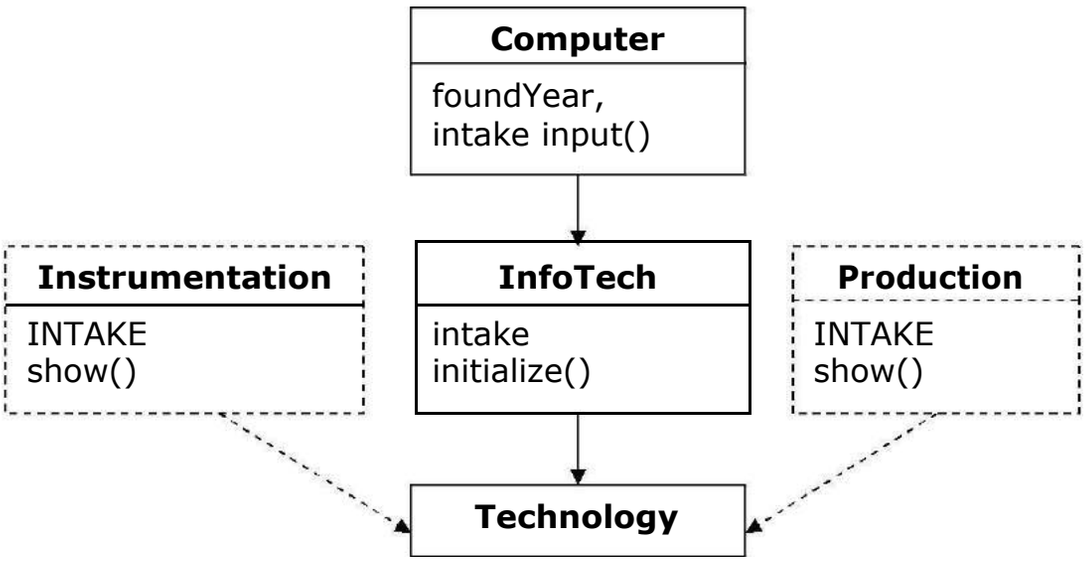
18. Implement the following inheritance:



19. Implement the following inheritance:



20. Implement the following inheritance:



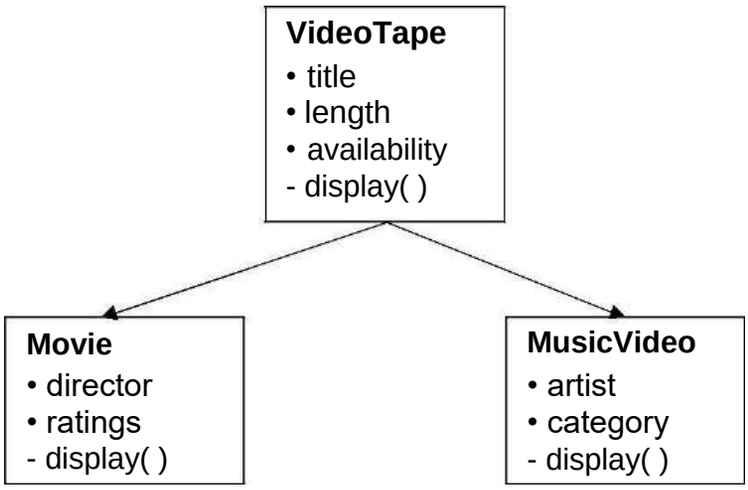
Prof. B.B. Jagadale
(Subject Teacher)

Practical No. 11

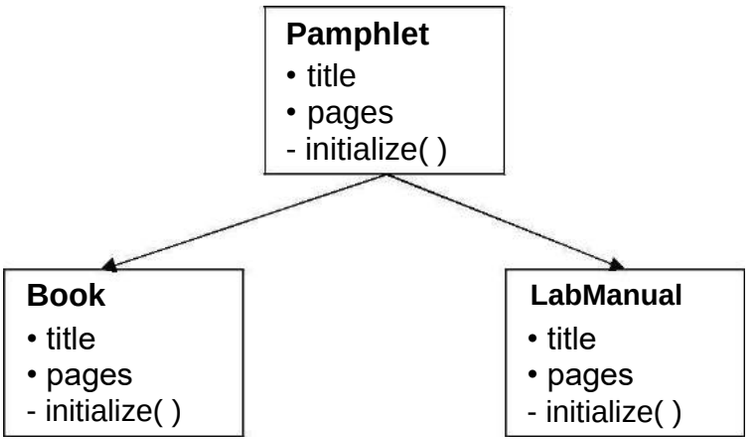
Subject: Java Programming **Class: TECSE**

Title: Method Overriding.

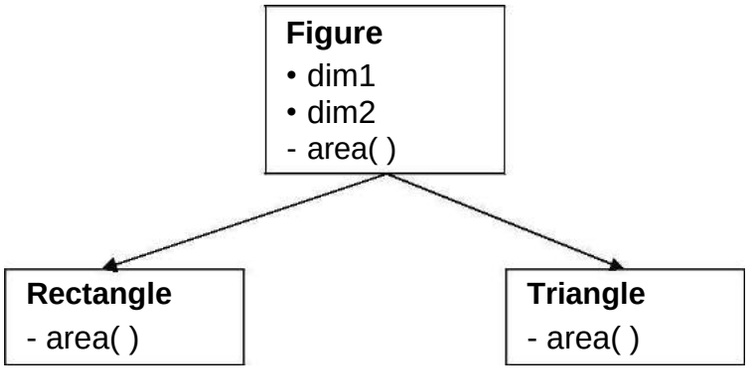
1. Declare class 'Ajax' having data members name, class and percentage. Inherit the class 'Bjax' from 'Ajax' which has data members total marks and name of college. Declare the overridden method input( ) and show( ) in both the classes. By creating the object of 'Bjax' call all the four methods to input and output these values.
2. Define a class 'Epnaňol' contains data members nation, language and population. Derive a class 'Franĉes' from it which contains data capital and continent. Both the classes are having overridden methods accept( ) and show( ). By creating the object of 'Franĉes' accept all these values and display the values.
3. Implement the following inheritance. Input the data using sub-class' constructors and display by calling all three versions of display( )(Winter'05).



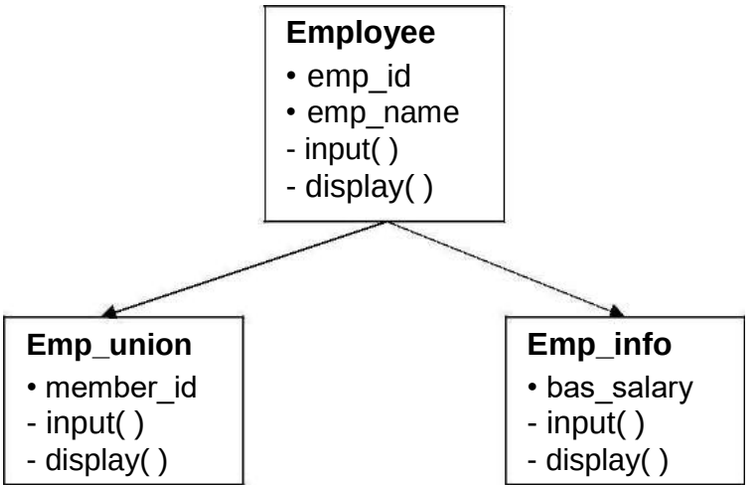
4. Implement following hierarchy and apply dynamic method dispatch to make a call to all methods. Display the data using sub-class' initialize( ) methods.



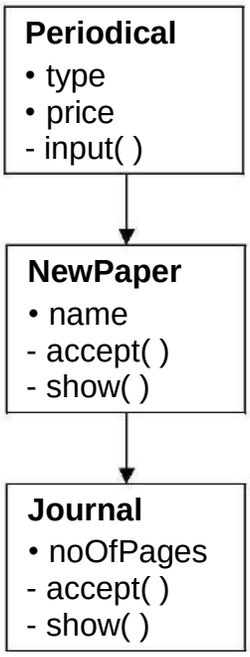
5. Implement following inheritance. Use constructors to initialize the objects. Create three objects of the respective classes and initialize their data members. Override method area() to find area of rectangle and triangle using dynamic method dispatch.



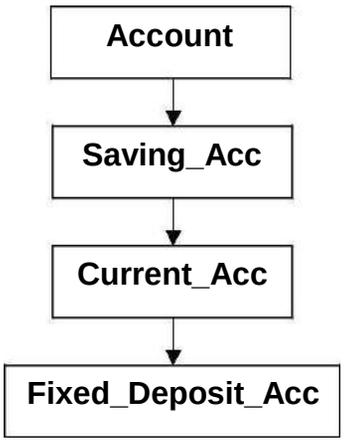
6. Define a class 'Student' having data members name and roll_no. Inherit class 'Teacher' from it to have data members name and subject. Derive one more class from 'Teacher' called 'Info'. All three classes contain the overridden method display to display respective information of the object. Input the data in object using constructors.
7. Observe the following figure of hierarchical inheritance to override the methods input( ) and display( ) methods. Initialize three objects of particular classes and display the values.



8. Create a class 'Land' contains variable 'scale' which is inherited in class 'Acres' and this 'Acres' class again inherited in class 'Hectors'. This multilevel hierarchy contains an overridden method 'convert( )'. Initialize the objects (variable: scale) using constructors. Make 'convert( )' as abstract in super class. Using dynamic method dispatch call convert method to convert inputted values in acres and hectors.
9. Implement following inheritance to input the values and show their contents. Input the values of periodical using input() and others using constructors. By applying dynamic method dispatch and abstract method display the inputted values.

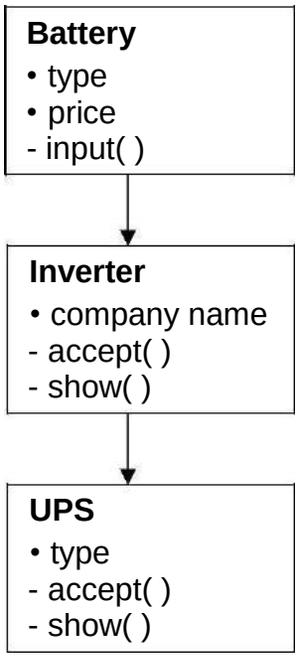


10. Implement the following inheritance. Assume suitable methods and data members in each class. Apply dynamic method dispatch appropriately to call the method in the program.

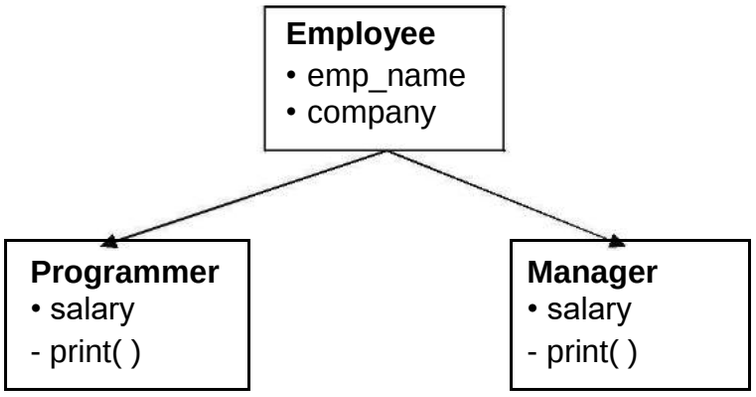


11. Create a class 'Cricket' which contains data members name of the country. Class 'Batsman' is inherited from 'Cricket' which contains data members such as name, runs scored and total innings. Class 'Bowler' is derived from 'Batsman' which has data members as name, runs given and overs bowled. All these classes contain two overridden methods input( ) and output( ) to input and output data. Define third method average( ) which will calculate the average of batsman and bowler separately. Apply dynamic method dispatch.
12. Define a class 'Fan' contains data member as company name and price. Derive class 'SealingFan' from it which has data members as speed and weight. Again derive one more class 'TableFan' from 'SealingFan' which contains data wattage. Initialize all data values using appropriate overridden method which is declared as abstract in super class. Override the display method to display inputted data values. By applying the dynamic method dispatch determine whose speed is faster.

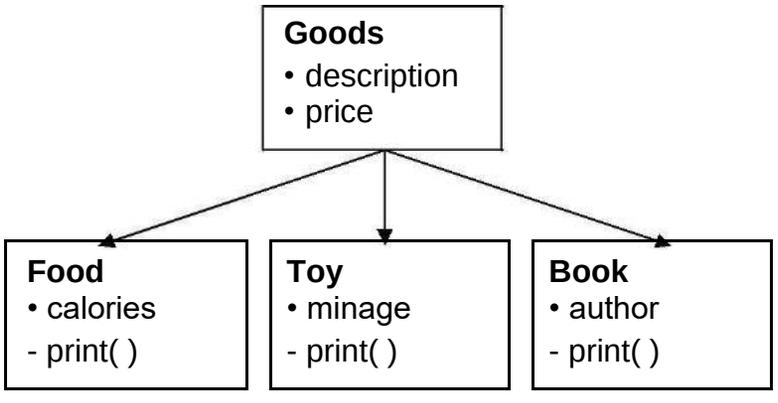
13. Create following inheritance to override the method show( ) and accept( ). Input and display the values of all data members of the objects. Apply dynamic method dispatch.



14. Create following inheritance to input values of 'Programmer' and 'Manager'. Override the method 'print() to output the values. Find whose salary is higher than each other.

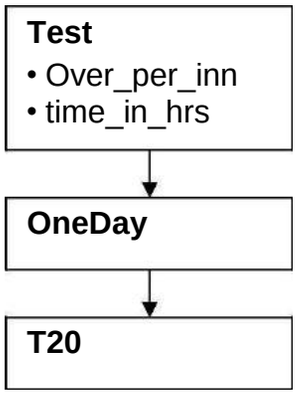


15. Consider the following hierarchy. Use method print() as abstract to print the initialized values written in the classes. Use concept of dynamic method dispatch.

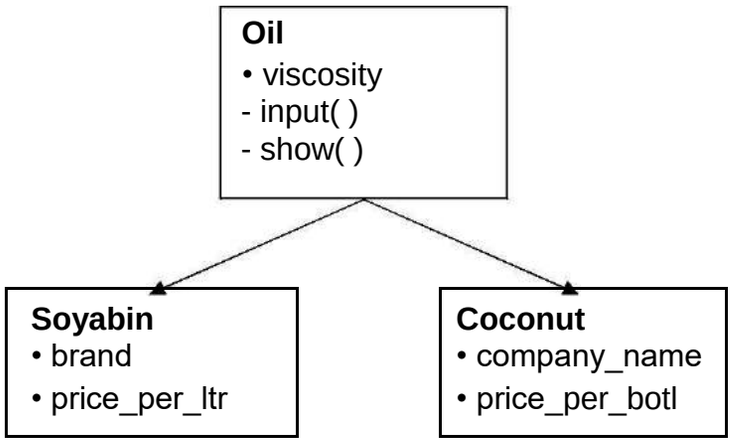


16. Create a class 'Cobol' inherited from 'Fortran' which is inherited from 'Pascal'. The class 'Cobol' contains data members such as name of creator, type and year of development. Define an overridden method displayIt() in all here classes. Input the values using constructor and display the data for programming language which is younger as per the year of development.

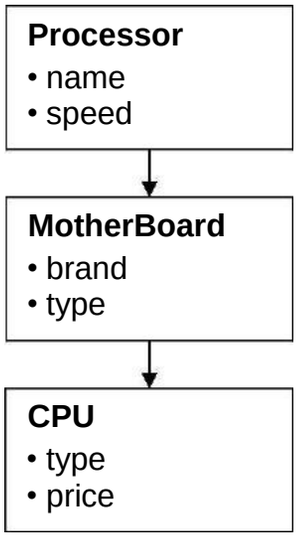
17. Implement the following inheritance. Apply dynamic method dispatch to create objects. Initialize the objects using constructors. Override the method show() to display all the values. Find whose time in hours is less than another two.



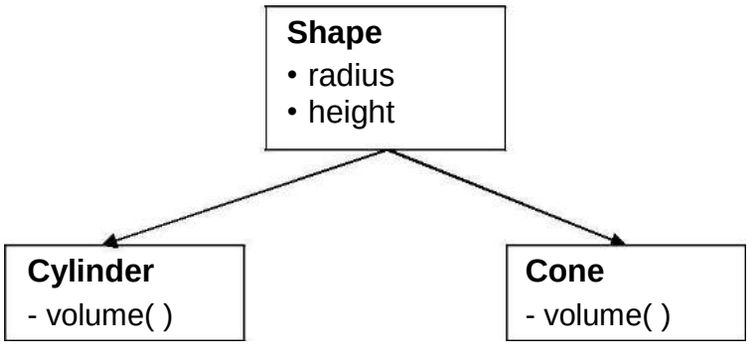
18. Implement the following inheritance to override the methods input( ) and display( ) and find whose viscosity is higher.



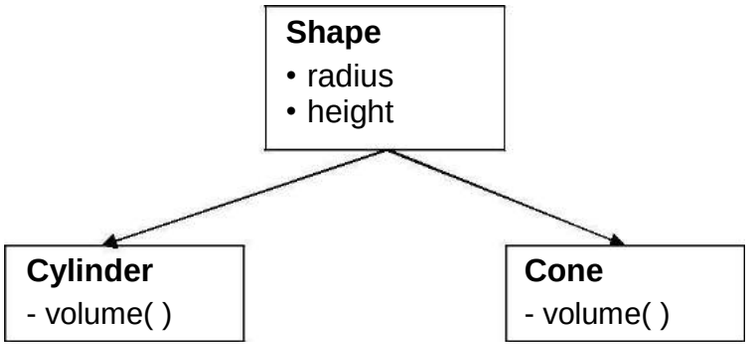
19. Implement the following inheritance by using suitable methods, making them abstract in super class. Input and display the data.



20. Implement the following inheritance to override the method volume() to find the volume of the cylinder and cone respectively.



21. Implement the following inheritance to override the method volume() to find the volume of the cylinder and cone respectively.



Prof.B. B. Jagadale
(Subject Teacher)

Practical No. 12

Subject: Java Programming **Class: TECSE**

Title: User Defined Exceptions

1. Accept any two numbers from command-line and find their addition. If the command-line arguments are not in number then throw CommandLineArgumentsException with message 'less arguments'.
2. Input a password from the keyboard and match it with your password. If it is not correct then throw AuthenticationFailureException.
3. Input any number from user, find its cube. Throw your named exception if this cube is not an even number.
4. Accept any three numbers from user and throw UnequalException is all three numbers are not equal. Display mismatched numbers as the message.
5. Create a positive number exception and throw it when result of your expression is positive. Display the message as the result of operation.
6. Input any string from the keyboard and throw an exception it this string does not contains 'A'.
7. Declare an integer array of ten numbers. Initialize it by taking user input. Throw an 'OverflowException' when the array reaches half of its final size.
8. Accept a character from the keyboard and throw NumberException when user entered the number as character.
9. Input a number from keyboard and throw an ArmStrongNumberException when the number is a Armstrong number.
10. Initialize two float variables from keyboard and throw 'TooSmallNumber Exception' when their resultant division is less than 0.000400f.
11. Use command-line parameters to input a string from keyboard and throw UpperCaseException when the string contains upper case letters.
12. Input any five integers from the keyboard and throw a MismatchException when any one of them is odd number.
13. Accept any long number from user and throw exception UnmatchException when its last character is an alphabet.
14. Input three numbers from the keyboard if their addition is a negative number then throw a NegativeNumberException.
15. Accept any string from keyboard and throw BooleanNumberException when this string is not a Boolean string.
16. Input two Boolean strings from command line and throw BooleanException when these strings are not equal.
17. Use data input stream to read a string and throw SpaceException when it contains a space in it. Unless print the string as it is.
18. Add any number of objects in a vector and throw the CapacityException when capacity exceeds ten.
19. Input any number of strings from command-line and throw EvenArguments Exception when the numbers of strings are not even unless print all the strings.
20. Accept a string and a number from keyboard. Match the string length with that number. If they are not equal throw string length exception. Else print length and the string.

Prof. Jagadale B.B.
(Subject Teacher)

Practical No. 13

Subject: Java Programming **Class: TECSE**

Title: Multithreading and Synchronization

1. Create two threads in your program. One will input and another will output the same integer data. Use Runnable interface.
2. Create a synchronized method to pass an integer as parameter to it. This method will print the digit by digit. Synchronize this method to print 756 7684 3865 by passing the separate string to separate threads.
3. Create two threads in your program. The thread class will contain a synchronized method. The parameter is passed to it as an integer. Print the integer in front of the name of the thread for 10 seconds.
4. Create a synchronized method to pass a string as parameter to it. This method will print the string *character by character.* Synchronize this method to print "The Java Programming Language" by passing the separate string to separate threads.
5. Create two different threads to print logarithm and square root of 1 to 200.
6. Input a string of any length once. Separate the upper case letters, lower case letters and digits by different threads.
7. Create three different threads that will contain input() method synchronized to input your name. The three threads will print "Hello *your-name*", "*your-name,* Good Bye" and "Welcome *your-name"* by calling their methods with the interval of 2 seconds each after input.
8. Create three different threads to print cosine value, cube and square root of 1 to 200 by implementing Runnable interface.
9. Create two different threads to print * and + one after each by interval of 400 milliseconds.
10. Create two different threads to print logarithm and square root of 1 to 200. Use Runnable interface.
11. Create three different threads to print your name in sequence with the interval of 600ms. Pass all three names to different threads.
12. Create two different threads to print all the odd numbers from 600 to 1 and 1 to 600 in sequence.
13. Create two different threads to create a synchronized method. This method will be passed an integer as parameter. Call this method for three threads with different parameters and find whether this number is odd or even with the interval of 500ms.
14. Create two threads so one thread will print ascending numbers whereas second thread will print descending numbers between sequence 1 to 15 numbers [Asked in Winter 2008].
15. Accept any long number from user using a method and check this number is Armstrong or not? Synchronize this method. Call the method for three threads.

16. Create two threads. One thread will print even numbers and another will print odd numbers from 1 to 10 [Asked in Summer 2004].

    The expected output is as below:

    Thread 1:1
    Thread 2:2
    Thread 1:3
    Thread 2:4
    .
    .
    .
    Thread 1:9
    Thread 2:10

17. Create a synchronized method to pass a string as parameters to it. This method will print the string *character by character.* Synchronize this method to print "The Java is Better than C++" by passing the separate string to separate threads.
18. Create three different threads to print the names of three countries by the interval of 2 seconds using a synchronized method.
19. Input an array of 10 integers. Separate the odd numbers, even numbers, positive numbers and negative numbers by different threads.
20. Accept any long number from user using a method and check this number is prime or not? Synchronize this method. Call the method for three threads.

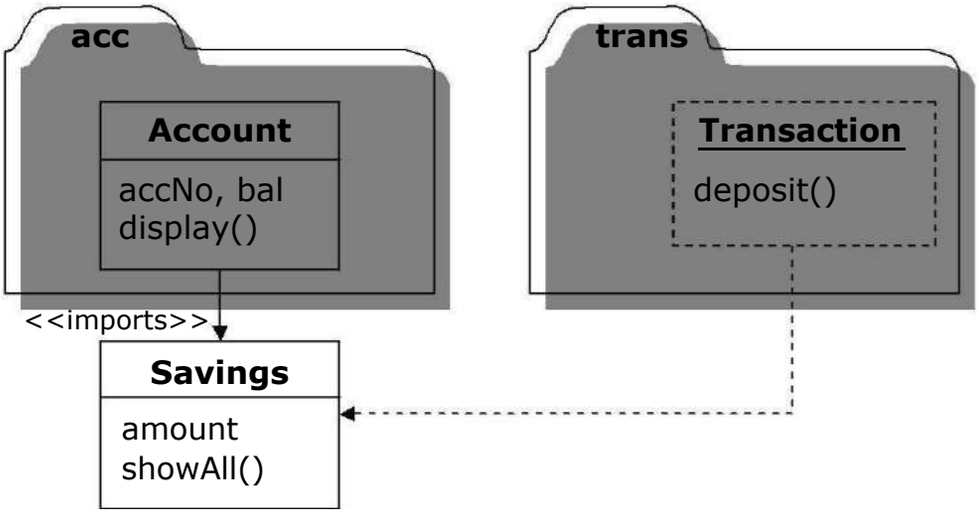
Prof. B. B. Jagadale
(Subject Teacher)

Practical No. 14

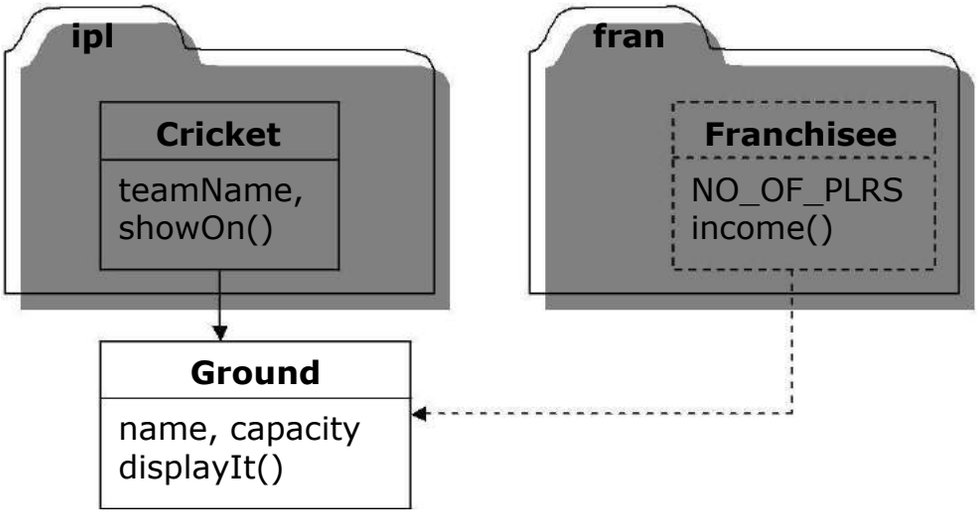
Subject: Java Programming **Class: TECSE**

Title: User-Defined Packages

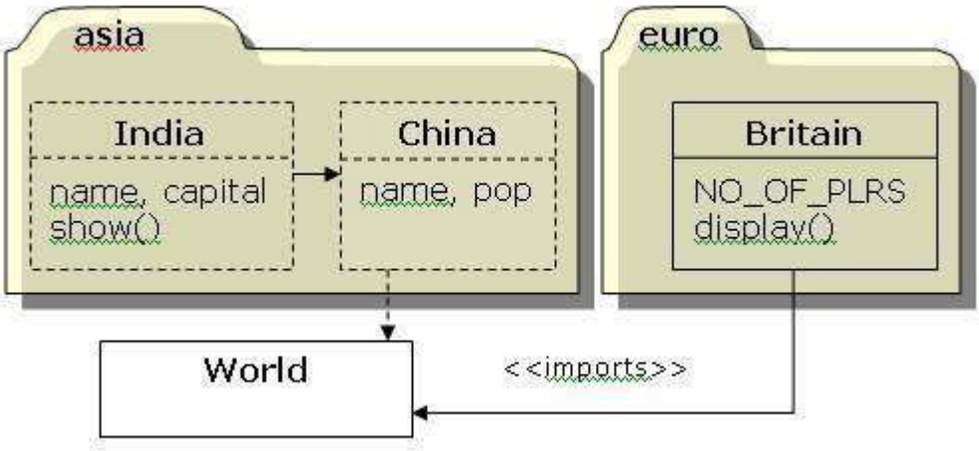
1. Implement the following diagram:



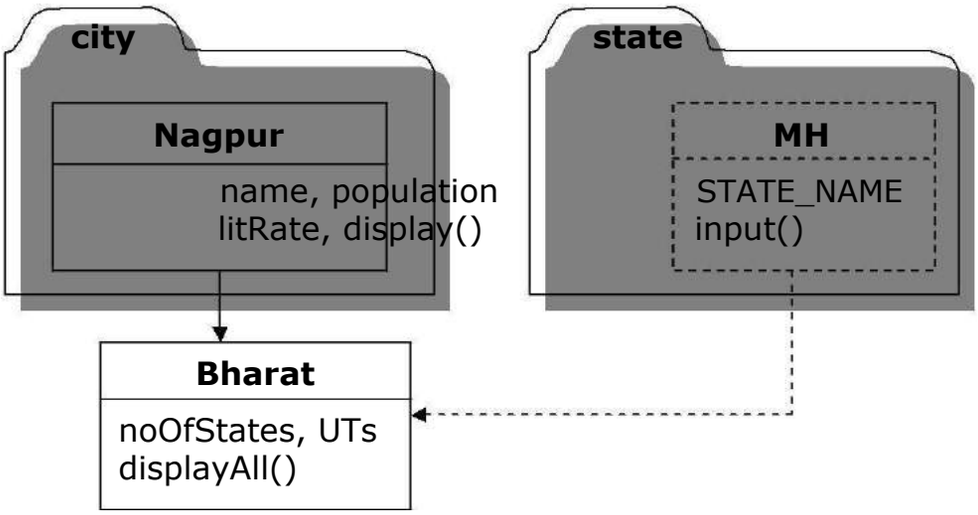
2. Implement the following diagram:



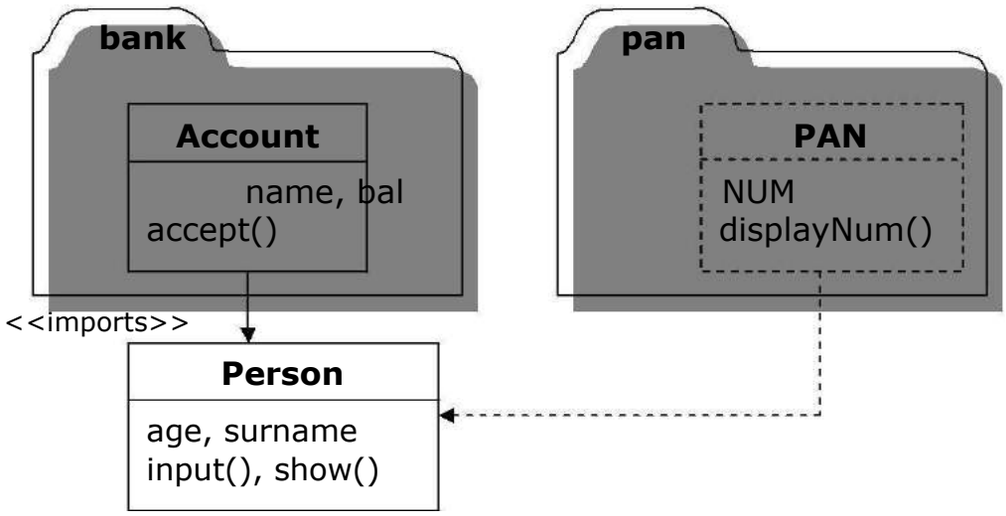
3. Implement the following diagram:



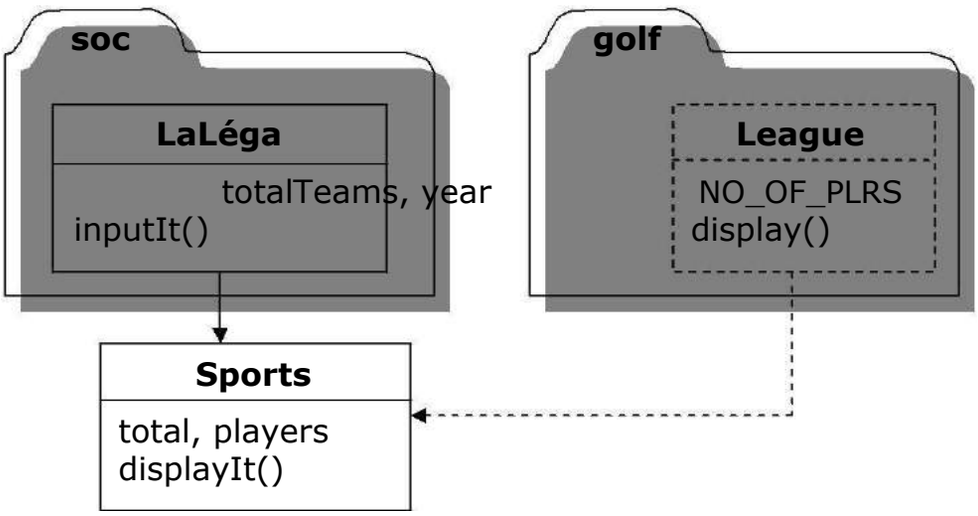
4. Implement the following diagram:



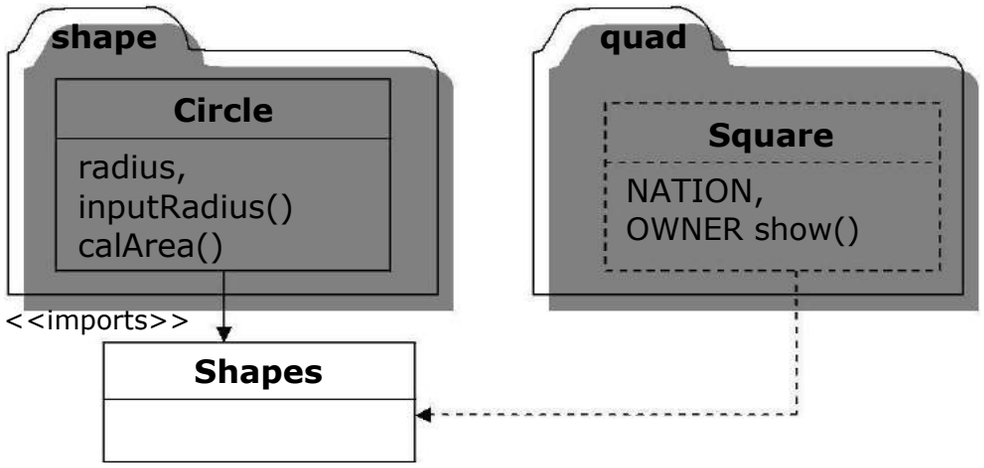
5. Implement the following diagram:



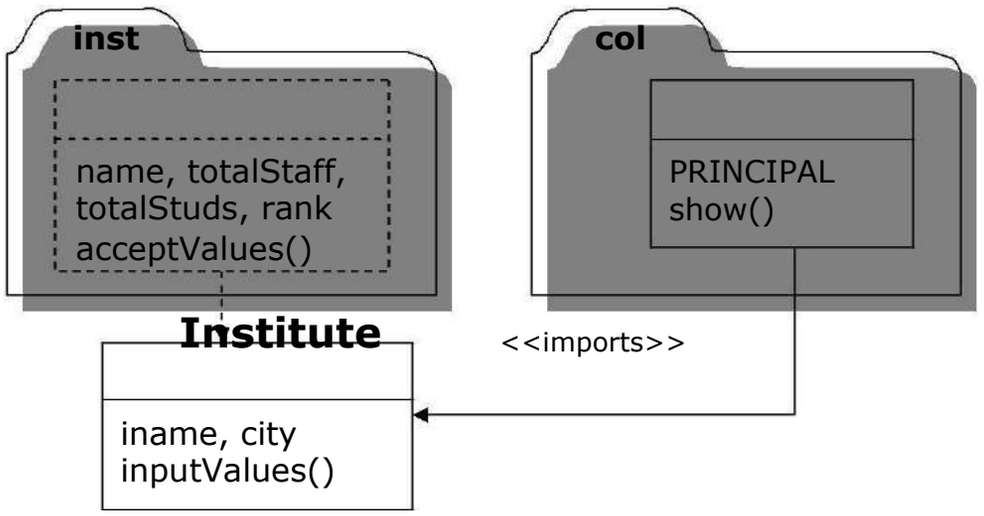
6. Implement the following diagram:



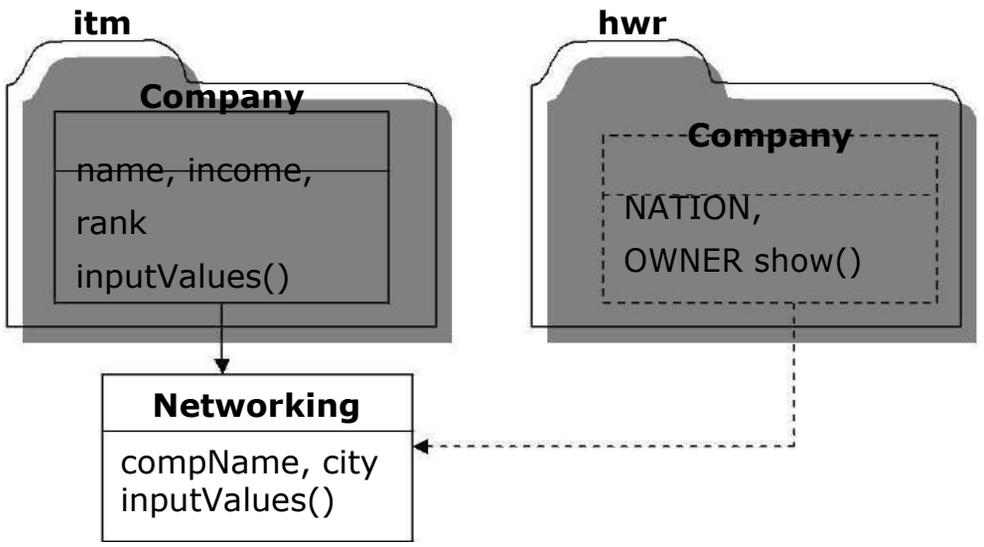
7. Implement the following diagram:



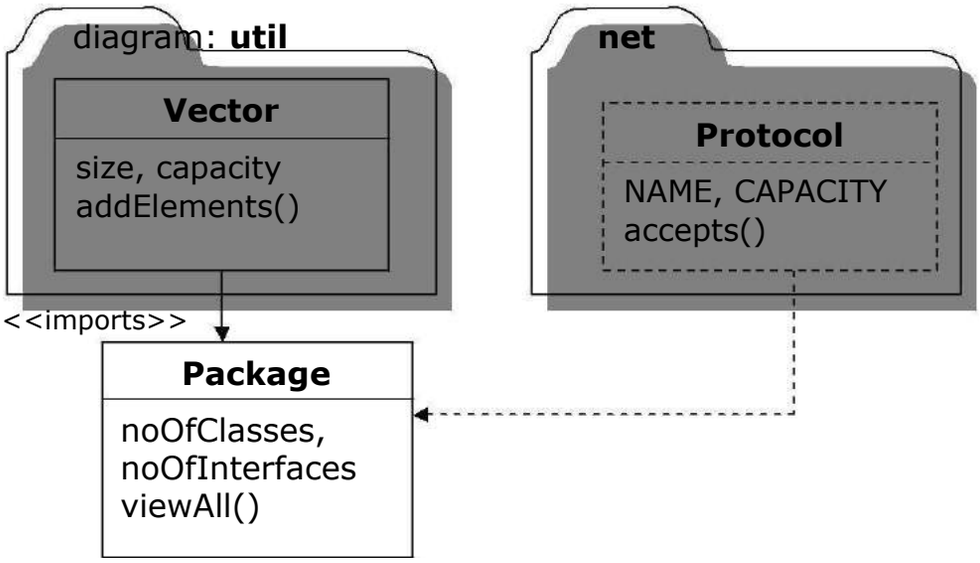
8. Implement the following diagram:



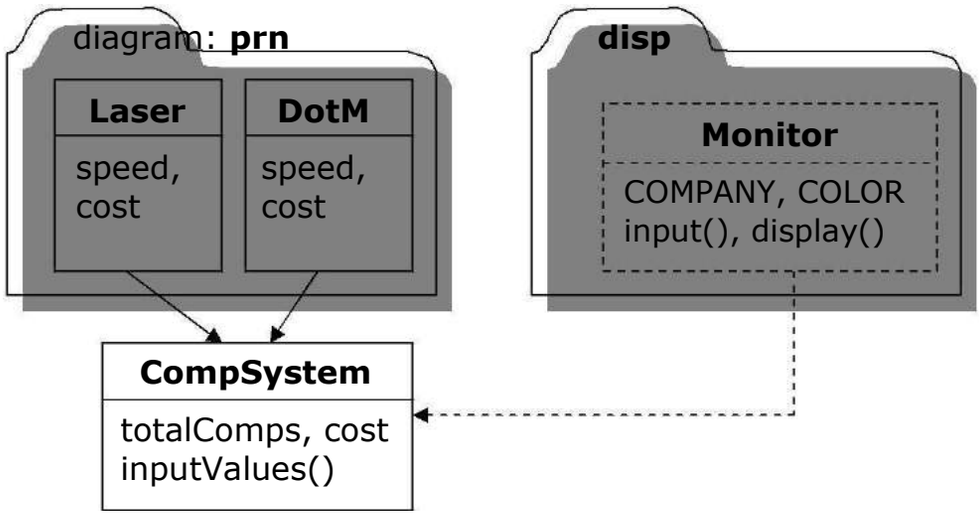
9. Implement the following diagram:



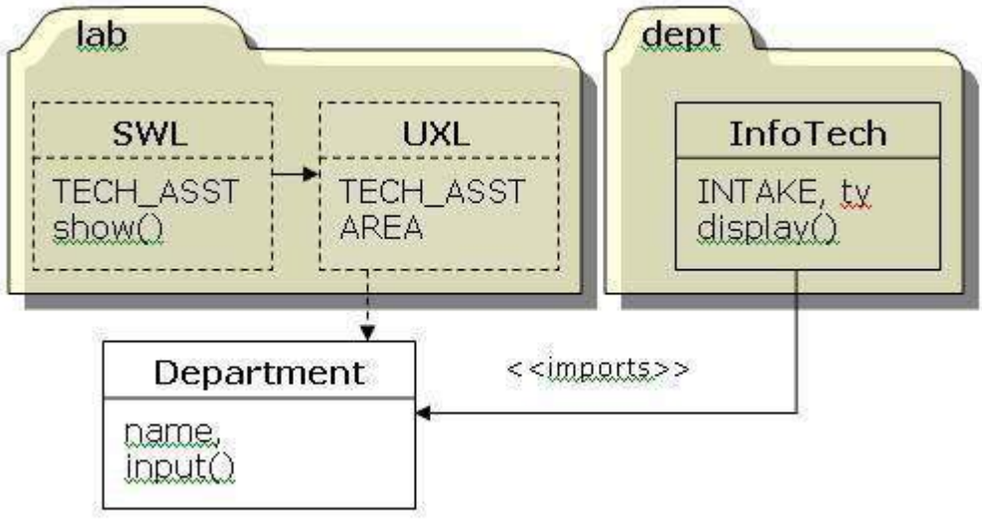
10. Implement the following



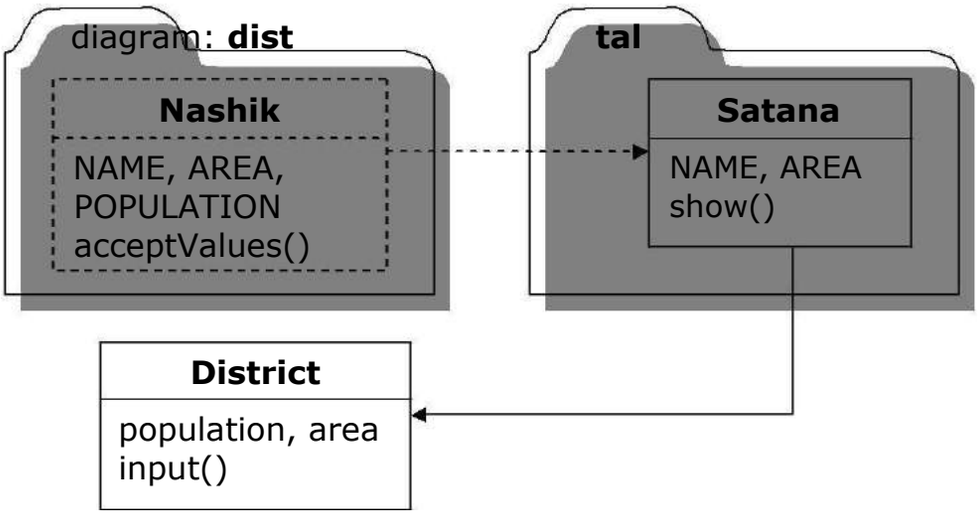
11. Implement the following



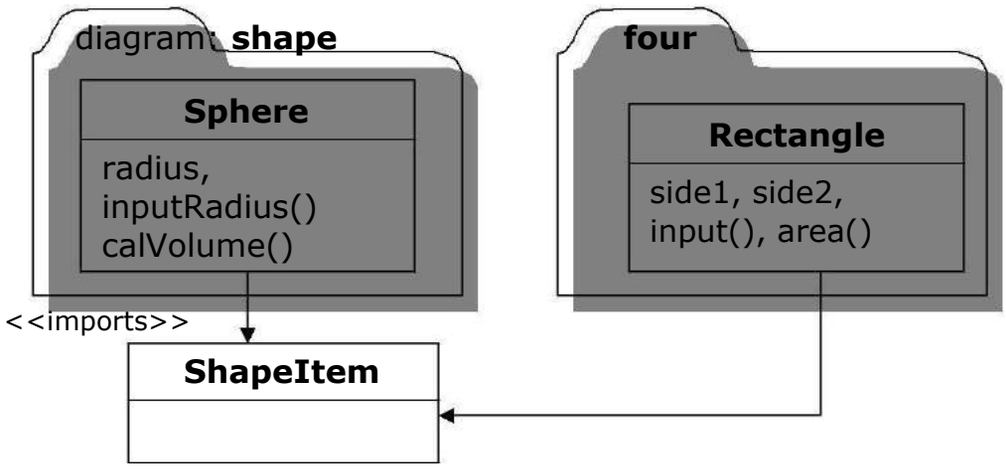
12. Implement the following diagram:



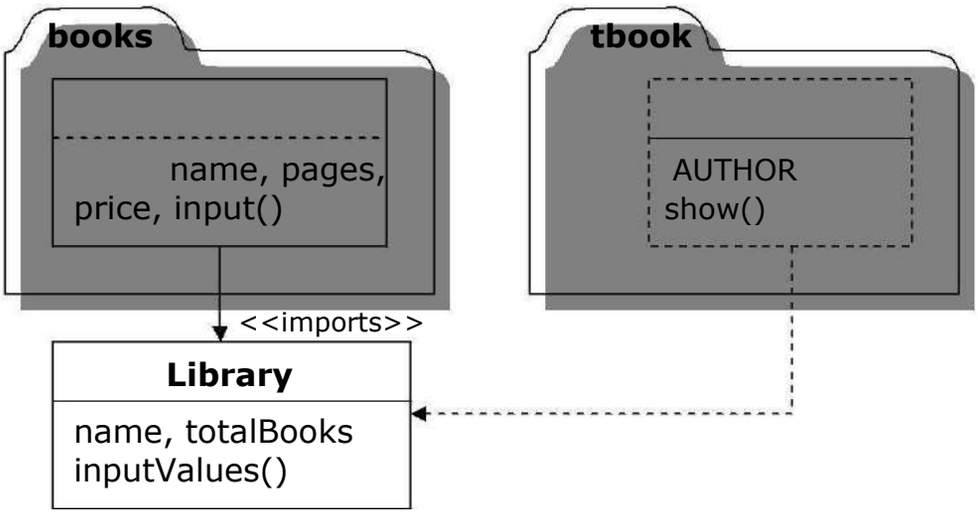
13. Implement the following



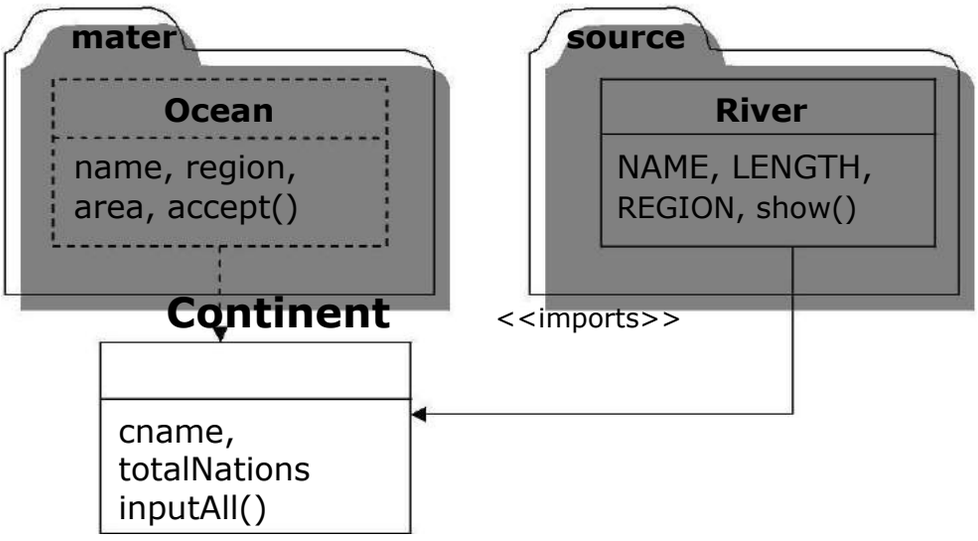
14. Implement the following



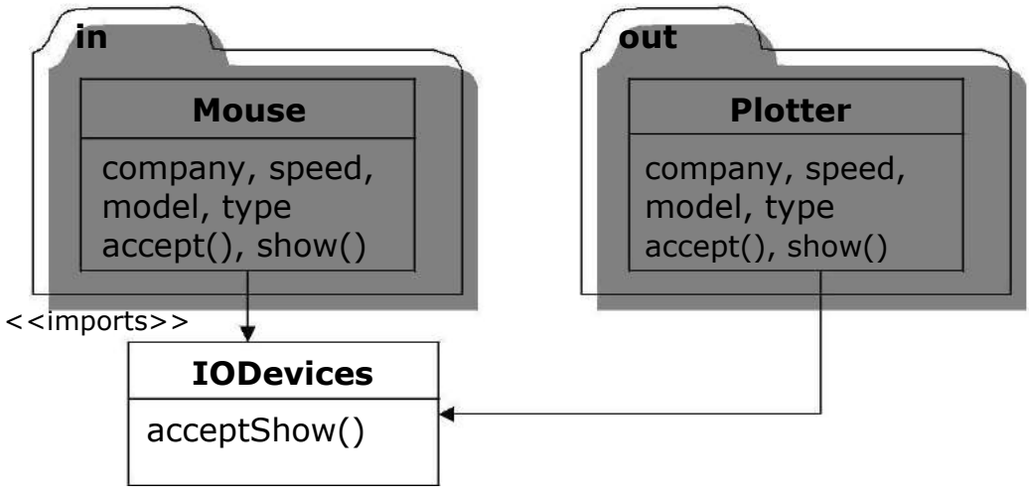
15. Implement the following diagram:



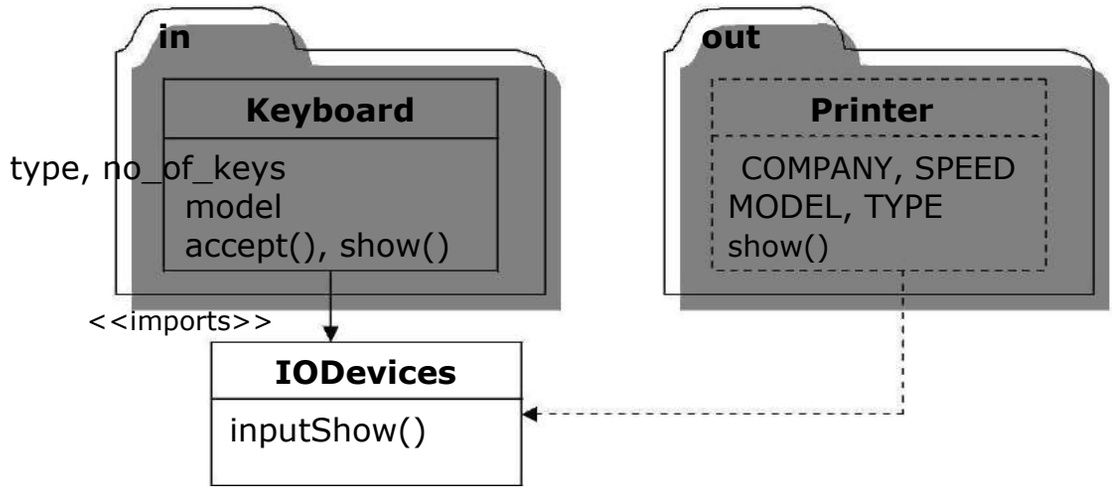
16. Implement the following diagram:



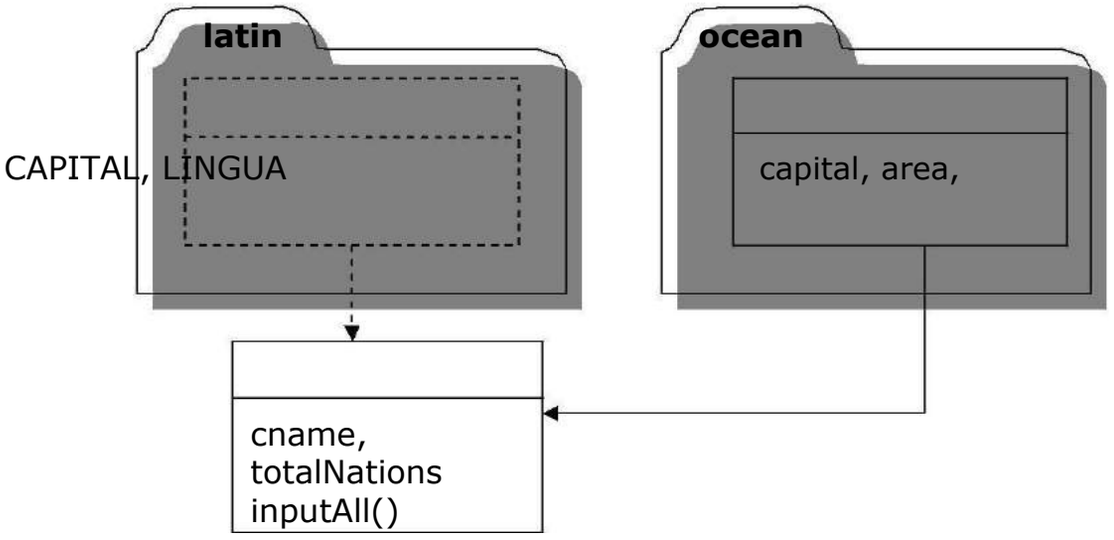
17. Implement the following diagram:



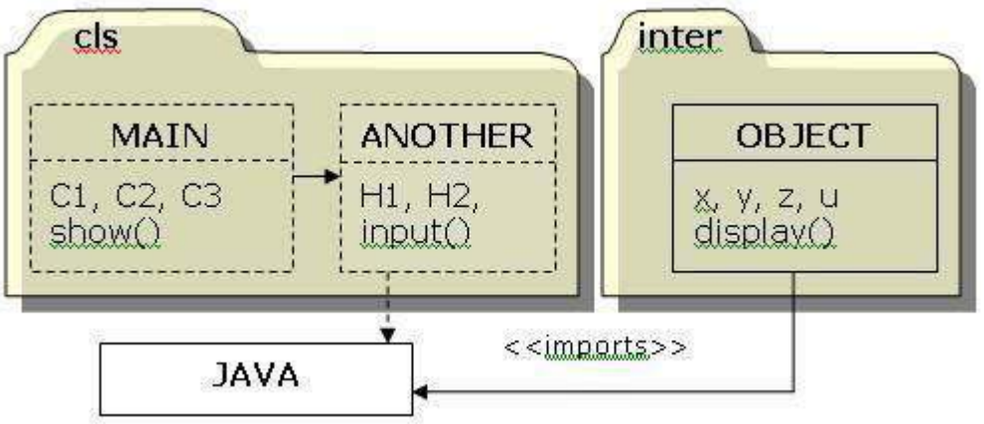
18. Implement the following diagram:



19. Implement the following diagram:



20. Implement the following diagram:



Prof. Jagadale B.B.
(Subject Teacher)

Practical No. 15

Subject: Java Programming **Class: TECSE**

Title: Design of Simple Applet

1. Accept the value of temperature in Celsius using param tag, convert it into Fahrenheit temperature. Display both values on the applet.
2. Create apple with pink background, red foreground. Display your name on the applet at 300, 200. Also display it on the status window.
3. Display the numbers from 10 to 1 on the applet with the delay of 200 milliseconds.
4. Display the name of your college on the applet (green background, gray foreground) with 1 seconds of interval between display of each character.
5. Create an applet to display all the vowels in English alphabets randomly on different positions. Use magenta background. Display this applet on the web page.
6. Input values of two ages and two names from the param tag and identify the name who is younger. Display it on the applet.
7. Create an applet with different fore and backgrounds. Accept any values from the param tag and display them on the status window.
8. Display all the odd numbers from 1 to 23 on the applet in diagonal fashion with pink background.
9. Input the distance in meters from param tag and convert it into centimeters. Display both values on applet as well as status window.
10. Accept the string “Welcome to Java Programming” from param tag and display each string separately on the applet with the interval of 500ms.
11. Display the names of colors “Black”, “Red”, “Magenta”, “Pink” on the applet with their respective colors using appropriate background color.
12. Input your name from the param tag and display it on applet as well as status window with the interval of 1 second. Use web page to view the applet.
13. Display all the numbers (1 to 10) as well as alphabets on the applet with different rows and different colors.
14. Create an applet with size 500 X 300. Resize it to (200, 200). Display your college“s name at 0,50 with red color and cyan background.
15. Input name of your college from the param tag and display it with red color with pink background on the applet. Also display it on the status window.
16. Accept names of five states and their capitals from the param tag and display them on the applet using web page. Use proper color combination.

17. Resize your applet with 400, 400 and display the string "I Love Java Programming" at the center of it. Display "Java" on the status window.
18. Print * on both the diagonals of the applet using any looping structure. Use appropriate color combination. Display it on the web page.
19. Create an applet with dimension 300, 300. Fill it with „–" on the whole screen. Apply suitable color combination. Display the status in the status window.
20. Use param tag to input values of x and y dimensions of the applet and resize it accordingly. Display your name at the centre of the applet by using proper color combination.

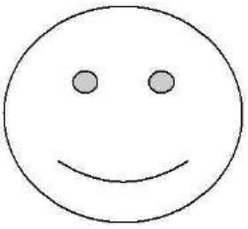
Prof. Jagadale B.B.
(Subject Teacher)

Practical No. 16

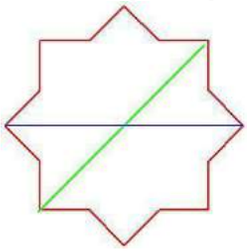
Subject: Java Programming **Class: TECSE**

Title: Design of Applet by Graphics class

1. Draw a flowchart to input two numbers from keyboard and find their addition.
2. Draw barchart to show your progress of percentage of marks in last four semesters and marks of tenth standard.
3. Draw five concentric ellipses with different colors. Draw a rectandle covering it.
4. Draw following diagram:

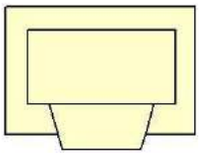


5. Draw following diagram:

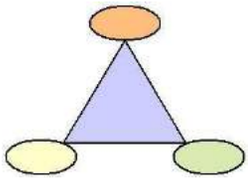


6. Draw a simple layout of computer includes monitor, keyboard and CPU.
7. Draw following diagram:

8. Draw a layout of football ground two side players with different colors and umpire with another color.
9. Draw following diagram:

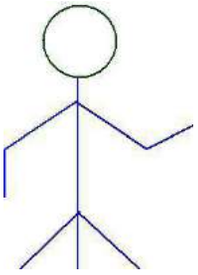


10. Draw following diagram:

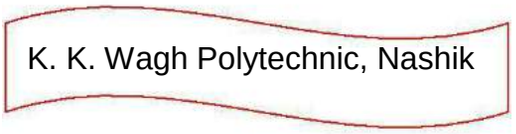


11. Draw the layout of your dream home.
12. Draw a wall clock showing current time.

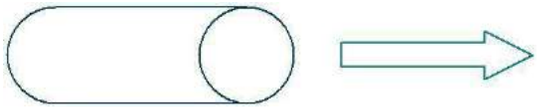
13. Draw a layout of cricket ground showing pitch and positions of all fielders with small circles.
14. Draw following diagram:



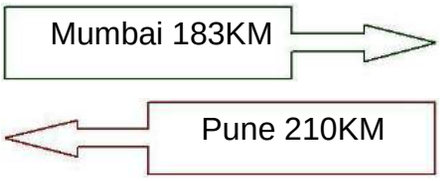
15. Draw the arial view layout of your college campus.
16. Draw follo diagram:



17. Draw following



18. Draw a lightening candle.
19. Draw following diagram:



20. Draw the outer transparent layout of glass cup.
21. Draw the flag of India.

Prof. Jagadale B.B.
(Subject Teacher)

JAVA PROGRAMMING

1. a. Display a text area, text field and a button (name is „calculate“) on the applet. The text area will input any number of lines from it. When we press the button named „calculate“, the text field will display total number of vowels present in the text of text area.

b. Write all the constructors of Text field, Text area and Button. Describe them.

2. a. Design a small scientific calculator which contains two text fields and four buttons. The text fields are for input and output. Buttons are named with „Log“, „Sine“, „Sqrt“ and „Tan“. When we input the number in input text field and press any particular button, the program will perform respective scientific calculation on the number and display it in the output text field. Display this calculator on the web page.

b. Write all the constructors of Text field, Text area and JButton. Describe them.

3. a. Create a text field, and three buttons on the applet. The buttons are named with „Arial Narrow”, “BOLD”, “ITALIC”. When we click on any of these buttons, the text entered from text field will get displayed on the applet (10, 150) positions with respective font type. Use the background color as pink.

b. Write all the constructors of JTextfield, JLabel and JButton. Describe them.

4. a. Create an event driven applet for handling mouse events. When you enter your mouse in the applet, the background color of the applet will get changed as you move the mouse on it. After exiting from the applet, the background color will be applied as „White“. After clicking on the applet the current color will be a stationary color of applet. This process will be continued until mouse is exited and entered.

b. Give description of all the methods of Mouse Listener interface. Describe how to handle Mouse Events.

5. a. Create a menu based program to show „file“ menu of Outlook Express.

b. Describe the process of creating Menus on the frame.

6. a. Create a menu based program to show „format“ and „tools“ menus of Microsoft Office PowerPoint.

b. Describe the process of creating menus on the frame.

7. a. Display a list and a choice control on applet. Display two buttons in between them named „Remove“ and „Add“. After pressing the buttons, the top item of the list will get inserted into choice deleting from it and vice versa.

b. Describe all the constructors of List and choice. Differentiate it.

8. a. Display two scroll bars, a rectangle and a circle on the applet. Use the values of the scrollbars to set the size of the rectangle and circle respectively.
b. Describe all the constructors of scroll bar. Give the meaning of constants defined by Scrollbar.

9. a. Display button on the applet with grid layout. When we click on it, a frame will be displayed with a scrollbar. After setting the maximum position to the scrollbar, the frame will be made invisible.
b. What do you mean by layout? Describe all the layouts available in Java"s AWT.

10. a. Create a frame with a list control, a button and a choice control. The list and choice both will contain names as „New", „Exit", „Open" and „Save". When we click on the button, the selected name of both list and choice must be „Exit" to make the frame invisible.
b. Enlist all the constructors of List, Button and Choice. Describe their uses.

11. a. Create two checkbox groups containing 10 items each which contains the numbers from 0 to 9 each. Display four buttons named „+", „–", „*" and „/". After clicking any of these buttons, the respective operations will be performed on the current selected radio button from the groups and the result will be displayed in the text area.
b. Describe the process of creating radio buttons in AWT. Describe all the constructors that are used in this process.

12. a. Display a list control on the applet. After we make the selection by double clicking the item name must be displayed in front of the string "Your Selection is:" on the applet.
b. What do you mean by a list control? Describe all the constructors of List.

13. a. Display the list control on the applet. It will contain five different names. When you select any of these names, it will be displayed on the status bar.

b. Describe all the methods that you have used in your program.

14. a. Display a choice control on the applet. After we make the selection by clicking the item name, its index must be displayed in front of the string "Your Selection is:" on the applet.
b. Describe the methods of choice controls. How to add any item to the choice?

15. a. Read the sent message from the client. The server will read the message and encode it its format. It will resend the same message by encoding it. Use encoding procedure as for example: "INDIA" encoded as "JMEJB".

b. Describe the process of creating client-server communication using TCP.

16. a. Use port number 13 for communication between client and server. Server will send its date and time to client after the request has received.
b. Describe the use of input stream and output stream classes in client server communication.

17. a. Establish a client-server communication to use port number 2041 as „analyze" port. That is, the client will send a string to local host on this port number. The server will analyze the string and respond by sending total

number of upper case letters, lower case letters, digits and special symbols in it.

b. What is the difference between the communication of client server using TCP and UDP? Describe in terms of Java.

18. a. Open the socket on server and client sides on port number 4233. The server will send the data continuously to client until ctrl+c is pressed.

b. Describe the process of creating UDP communication.

19. a. Establish a client server communication through port 5525. The communication will be completed after the server sends message: "END".

b. Describe the use of datagram packet and datagram socket classes with their constructors.

20. a. Create the array of following URLs and identify the type of the content that each contents:
http://presidentofindia.nic.in/speeches.html
http://www.telugufm.com/Modules/music/Actor.aspx
http://www.nineplanets.org/jupiter.html
http://www.priu.gov.lk/Human_Rights.html
http://www.lib.utexas.edu/taro/utcah/00204.xml

b. Describe the process of creating URLConnection.

21. a. Use getHeaderField( ) method to identify the content type, content length, content encoding for the following URL:
http://en.wikipedia.org/wiki/Central_processing_unit

b. Enlist and describe the methods of URL class.

22. a. Download the following webpage using and display its contents on the screen.
http://presidentofindia.nic.in/speeches.html

b. Describe the methods defined by URLConnection class.

23. a. Send the data from server to client using UDP. The client will analyze this data by finding total number of characters, digits and special symbols.

b. What are advantages of creating UDP communication using a Java program? Describe constructors of datagram packet.

24. a. Send the data from sender to the receiver using datagrams. When the data is received at receiver"s end does not starts with upper-case letter it will output:- data reception failed….!

b. Describe the process of creating UDP communication using Java.

25. a. Identify and display the information of all the newspapers in Marathi published in Nashik region for following database:
Database:
Columns: Serial Number, Newspaper name, Language, Region, Circulation

b. Describe the use of Connection, Statement and ResultSet.

26. Delete the information of all the players who has played matches less than 50 and display remaining information. Identify the count also.
Database:
Columns: Number, Player Name, Matches, Runs, Average, 100s, 50s.

b. Describe the use of Connection, Statement and ResultSet.

27. Display the information of all the books whose pages are in between 200 and 400.

Database:
Columns: Book name, pages, author, publication, language
DSN: Books Entries: 10

b. Describe the process of creating JDBC-ODBC Bridge.

28. a. Identify who has got marks in the range of „x“ and „y“ in the class. Initialize „x“ and „y“ from user.
Database:
Columns: Roll, Name, Age, Class, Marks

b. How to establish and use database communication using JDBC application? Describe.

29. a. Display names of players who have scored both 100s and 50s more than „d“. Initialize value of „d“ from keyboard.
Database:
Columns: Number, Player Name, Matches, Runs, Average, 100s, 50s.

b. Describe the use of PreparedStatement.

30. Create a menu based program to show the information of all the mobiles as per the descending or ascending order of cost.
Database:
Columns: mobile Company, Serial Number, Cost, Memory capacity

b. Describe the use of Callable Statement.

31. a. Swing:- Change the background color of the applet as per the selection made from radio button items. The status bar will display the current selected color.

b. What differences you felt in between AWT and Swing using programming?

32. a. Swing:- Display a text field, a button and a check box on the applet. When you click onto the button, the text from text field will get set as the label on the check box.

b. What are the extra features that are added by swing over AWT?

33. a. Swing:- Display four different text boxes and buttons on the applet. Every text box is associated with appropriate button. When you click any of these buttons, the text of the associated text box will get displayed on the status bar.

b. Describe the use of Content pane, button, label and textfield of swing.

34. a. Create following tabbed pane (Find and Replace):
Tab1- Find (Label: “Find what”, a Text field, two Buttons: “OK”, “Cancel”)
Tab2- Replace (two Labels: “Find what”, “Replace with”, two Buttons: “Replace All”, “Cancel”)
Tab3- Goto (A Combo box (6 items), A text field and two buttons)
Place tabs on the top. Use „wrap tab policy“

b. Describe the process of creating tabbed pane with the help of a flowchart.

35. a. Create following tabbed pane (Options-Spelling and Grammar):
Tab1- Spelling (seven check boxes and a button)
Tab2- Grammar (Four check boxes, a combo box and a button) Tab3- Proofing tool (A Button)
Place tabs on the left. Use „wrap tab policy“.

b. Describe the process of creating tabbed pane with the help of a flowchart.

36. a. Swing:- Display two combo boxes on the applet. They will contain the numbers associated with them. The text field near to them will contain the multiplication of numbers from both selected combo boxes. Apply border layout and scroll pane for both scrollbars.

b. Describe the process of creating scroll pane with the help of a flowchart.

37. a. Identify the hierarchy and show the following items in the form of tree by including the sub-tree elements:
„JComponent", „Applet", „Window", „Panel", „Object", „Container", „Jpanel", „Frame", „JTree"

b. Describe the process of creating tree with the help of a flowchart.

38. a. Create the binary search tree for the following elements:
„V", „G", „T", „N", „Z", „W", „J", „Q", „A", „F"

b. Describe the process of creating tree with the help of a flowchart

39. a. Create the view of any tree topology of network which is having the following network computer names:
„UNIX39", „SWL02", „UNIX50", „UNIX63", „SWL13", „ITDEPT-SERVER", „UNIX59", „SWL20", „SWL05", „UNIX60".

b. Describe the process of creating tree with the help of a flowchart

40. a. Open your computer"s windows explorer and create the hierarchy shown for first three directories/folders.

b. What do you mean by a tree? Describe all the classes and interfaces associated with tree.

Prof.B.B. Jagadale

(Subject Teacher)

Question Bank:

Title: Program based on TextField, TextArea, Label and Button

1. Display a text area, text field and a button (name is 'calculate') on the applet. The text area will input any number of lines from it. When we press the button named 'calculate', the text field will display total number of vowels present in the text of text area.
2. Display three text fields on the applet. Also display a button below to all. We have to insert the integer values from three text fields. After pressing the button, the background color of the applet will be changed as per the RGB combination of the values given in the text field.
3. Display two text areas near to each other on the applet. A button named 'Reverse' is displayed below to them. We need to insert the text in the first text area. After pressing the button second text area will display the reverse string of the first text area which is non-editable.
4. Display five different buttons on the applet with the names of colors as Red, Yellow, Pink, Green and Cyan. When we press any button, the background color of applet will get changed as per the name present on the button.
5. Design a small scientific calculator which contains two text fields and four buttons. The text fields are for input and output. Buttons are named with 'Log', 'Sine', 'Sqrt' and 'Tan'. When we input the number in input text field and press any particular button, the program will perform respective scientific calculation on the number and display it in the output text field. Display this calculator on the web page.
6. Display a text field, a text field near to each other on the applet. Also display a button named 'show' below to them. The text field will show only character 'X', when any text is inserted in it. After pressing the button the actual text will get displayed in the text area which is non-editable.
7. Display two text fields near to each other on the applet. Also display a button named 'show' below to them. The first text area will show only character '@', when any text is inserted in it. After pressing the button the actual text will get displayed in the second text field in reverse order, which is non-editable.
8. Display two text fields and three different buttons in between them. Buttons are named with 'IS ODD?', 'IS PRIME?' and 'ISNEGATIVE?'. Enter the number in the first text field and after pressing button, second text field will display the result as per the text present on them. That is, check whether the number is prime, odd or negative. Display the applet on the web page.
9. Display a text area, a text field and a button on the applet. We need to insert the text in the text area. When we select some text in the text area and press the button, then the selected text will get displayed in the text field.
10. Display two text fields and a button in a column on the applet. The button will contain the label 'ACTION'. We need to insert a number in first text field and the 'action' in the second text field. After pressing the button, the action given in the second text field will get performed on the number from the first text field. The actions given are – 'ISODD', 'ISNEGATIVE', 'CUBE', 'SQRT',

'SQUARE', 'EXIT' etc. The second text field will find particular result of the first number.

11. Create a number systems converter calculator to input a number in decimal number system and display its binary, octal and hexadecimal equivalents. Make the use of text fields and buttons to input and output.

12. Create a digital operation calculator to input two numbers and perform the operations of AND, OR, NOT and EX-OR on them. Make the use of text fields and buttons to input and output.

13. Display two text areas on applet. As soon as we input the text in first text field, the upper case letter of the same text will be printed in the second text field.

14. Design an applet to input the information required for taking admission in the college. It will contain a 'Submit' button. When it is clocked all the related information must get stored in a file. Make the use of labels and text boxes.

15. Create an applet to display the text areas with various values. Implement the button to perform certain operations. When we move our mouse inside the applet the text areas must get disabled by this action. So in order to activate the input from text fields we must keep the mouse outside the applet window.

16. Display six different text boxes on the applet. Input the numbers from them in the program. Determine the largest number among them. Use text box to display the largest number. Implement the action listener to take input and output. When all numbers entered are same, the applet must terminate its execution.

17. Use a label, a text field and a button to show it on the applet. When we press the button, a circle should be displayed on the applet area with the specified radius entered through the text field.

18. Display four text fields named– left, top, right and bottom. These will be used to enter the respective coordinates of the rectangle. When we press the button added on the, applet the rectangle must be drawn with corresponding coordinates on applet display area. If the negative values are entered in the text fields, display the error message on the applet.

19. Create a text area on the applet to input a paragraph of English text. Whenever we press the button added on the applet, the text field should display total number of lines entered in the text area. (Here the applet contains four controls: Label, TextArea, TextField and Button).

20. Display a text field and ten buttons labeling from 1 to 10 digits. When we press the respective buttons, number should be produced in the text field. Same implementation is done as in pocket calculator or windows calculator.

Question Bank:

Title: Program based on Different Layouts

1. Display two text areas near to each other on the applet. A button named 'Reverse' is displayed below to them. We need to insert the text in the first text area. After pressing the button second text area will display the reverse string of the first text area which is non-editable. Apply border layout.
2. Display five different buttons on the applet with the names of colors as Red, Yellow, Pink, Green and Cyan. When we press any button, the background color of applet will get changed as per the name present on the button. Apply grid layout and insets of (50, 50, 50, 60).
3. Use a label, a text field and a button to show it on the applet. When we press the button, a circle should be displayed on the applet area with the specified radius entered through the text field. Apply flow layout left.
4. Display four text fields named– left, top, right and bottom. These will be used to enter the respective coordinates of the rectangle. When we press the button added on the, applet the rectangle must be drawn with corresponding coordinates on applet display area. If the negative values are entered in the text fields, display the error message on the applet. Apply flow layout right.
5. Create a text area on the applet to input a paragraph of English text. Whenever we press the button added on the applet, the text field should display total number of lines entered in the text area. (Here the applet contains four controls: Label, TextArea, TextField and Button). Apply border layout.
6. Design a small scientific calculator which contains two text fields and four buttons. The text fields are for input and output. Buttons are named with 'Log', 'Sine', 'Sqrt' and 'Tan'. When we input the number in input text field and press any particular button, the program will perform respective scientific calculation on the number and display it in the output text field. Display this calculator on the web page. Apply border layout.
7. Display a text area, text field and a button (name is 'calculate') on the applet. The text area will input any number of lines from it. When we press the button named 'calculate', the text field will display total number of vowels present in the text of text area. Apply grid layout.
8. Display three text fields on the applet. Also display a button below to all. We have to insert the integer values from three text fields. After pressing the button, the background color of the applet will be changed as per the RGB combination of the values given in the text field. Apply grid layout and insets of (10, 20, 30, 40).
9. Display a text field, a text field near to each other on the applet. Also display a button named 'show' below to them. The text field will show only character 'X', when any text is inserted in it. After pressing the button the actual text will get displayed in the text area which is non-editable. Apply flow layout left.
10. Display two text fields and a button in a column on the applet. The button will contain the label 'ACTION'. We need to insert a number in first text field and the 'action' in the second text field. After pressing the button, the action given in the second text field will get performed on the number from the first

text field. The actions given are – 'ISODD', 'ISNEGATIVE', 'CUBE', 'SQRT', 'SQUARE', 'EXIT' etc. The second text field will find particular result of the first number. Apply grid layout.

11. Create a number systems converter calculator to input a number in decimal number system and display its binary, octal and hexadecimal equivalents. Make the use of text fields and buttons to input and output. Apply grid layout.

12. Create a digital operation calculator to input two numbers and perform the operations of AND, OR, NOT and EX-OR on them. Make the use of text fields and buttons to input and output. Apply border layout and insets of (20, 20, 30, 40).

13. Display two text areas on applet. As soon as we input the text in first text field, the upper case letter of the same text will be printed in the second text field. Apply grid layout.

14. Design an applet to input the information required for taking admission in the college. It will contain a 'Submit' button. When it is clocked all the related information must get stored in a file. Make the use of labels and text boxes. Apply flow layout left.

15. Create an applet to display the text areas with various values. Implement the button to perform certain operations. When we move our mouse inside the applet the text areas must get disabled by this action. So in order to activate the input from text fields we must keep the mouse outside the applet window. Apply insets of (10, 20, 50, 40).

16. Display six different text boxes on the applet. Input the numbers from them in the program. Determine the largest number among them. Use text box to display the largest number. Implement the action listener to take input and output. When all numbers entered are same, the applet must terminate its execution. Apply grid layout.

17. Display a text field and ten buttons labeling from 1 to 10 digits. When we press the respective buttons, number should be produced in the text field. Same implementation is done as in pocket calculator or windows calculator. Apply grid layout.

18. Display two text fields near to each other on the applet. Also display a button named 'show' below to them. The first text area will show only character '@', when any text is inserted in it. After pressing the button the actual text will get displayed in the second text field in reverse order, which is non-editable. Apply border layout.

19. Display two text fields and three different buttons in between them. Buttons are named with 'IS ODD?', 'IS PRIME?' and 'ISNEGATIVE?'. Enter the number in the first text field and after pressing button, second text field will display the result as per the text present on them. That is, check whether the number is prime, odd or negative. Display the applet on the web page. Apply grid layout.

20. Display a text area, a text field and a button on the applet. We need to insert the text in the text area. When we select some text in the text area and press the button, then the selected text will get displayed in the text field. Apply flow layout left.

Question Bank:
Title: Program based on Fonts and Colors

1. Create an event driven applet program which will change the background color of applet when the mouse is entered in the applet or clicked inside the applet.
2. Create a text field, and button on the applet. The text field will accept the name of the color. After pressing the button the background color of applet will get changed. If the color does not match, it will give error.
3. Accept a string and its position from text field and display it on specified position on the applet with „Verdana“, Bold, Italic, 23.
4. Display a text area and three buttons on the applet named “Times”, “Courier” and “Arial Narrow”. When we click on any of these buttons, the string entered from text area will get displayed in respective font.
5. Display a text area and six buttons on the applet named “Red”, “Green” “Pink”, “Magenta”, “Cyan”, and “Blue”. When we click on any of these buttons, the string entered from text area will get displayed in respective color with gray background, Times New Roman, 20, Underlined font and Flow layout-left.
6. Create an applet to display the text field. When we enter the mouse over the applet the string entered in the text field will get displayed in “Arial”, 25, Plain font at position 50,150. And When we exit the mouse from the applet the string entered in the text field will get displayed in “Arial Narrow”, 20, Bold font at position 50,150.
7. Create an event driven applet program to display your name at the center with Red background, RGB (200,145,230) foreground, Book Antigua, 19, BOLD, ITALIC. When we enter mouse in applet, the displayed text will be changed to RGB (100,20,190) background, RGB (200,120,250) foreground, Book Antigua, 25, ITALIC, UNDERLINED. When we click mouse in applet, the displayed text will be changed to Black background, Gray foreground, Verdana, 22, Bold, ITALIC, UNDERLINED.
8. Create an event driven applet program to display your name at the center with Light Red background, RGB (150,245,10) foreground, Arial, 20, BOLD, ITALIC. When we press „F1“ key on applet, the displayed text will be changed to RGB (10,220,90) background, RGB (30,220,150) foreground, Times, 35, ITALIC, UNDERLINED. When we press „F2“ key in applet, the displayed text will be changed to Gray background, Light Blue foreground, Arial Narrow, 25, Bold, ITALIC, UNDERLINED.
9. Display a text area on applet with font „Courier New“. When we press the button present on applet the same text will get displayed in another text area with „Times New Roman“ font. Set the foreground color as RGB (120,230,200).
10. Create an event driven applet to display your name on it. When we click on the applet display area, the color and font size of the text will get changed each time differently. Use „Times New Roman“ as default font.
11. Display a text field and a button on the applet. When we enter the text in the text field it will get displayed on applet at (100,200) with Courier New, 26, BOLD, ITALIC. After each click on the button, the size of the font will get changed.

12. Display a text field and a button on the applet. When we enter the text in the text field and click on the button, the text field"s text will get displayed on the applet with different fonts, font sizes, and font styles on different positions.
13. Display name of your college on the applet with „Verdana", 18, Bold, Red. When you drag the mouse on the applet, font size of the text will get incremented and the background color will get changed.
14. Create an applet to display the Indian flag on the applet. Display the text „Indian Flag" below it with "Times New Roman", 25, ITALIC.
15. Create a text field, and three buttons on the applet. The buttons are named with „Arial Narrow", "BOLD", "ITALIC". When we click on any of these buttons, the text entered from text field will get displayed on the applet (10, 150) positions with respective font type. Use the background color as pink.
16. Create an event driven applet for handling mouse events. When you enter your mouse in the applet, the background color of the applet will get changed as you move the mouse on it. After exiting from the applet, the background color will be applied as „White". After clicking on the applet the current color will be a stationary color of applet. This process will be continued until mouse is exited and entered.
17. Display four different buttons and a text field on applet. Use insets positions as (20, 20, 20, 20). Apply background color as „Light Red". Use border layout to set the text field at the center. The buttons are named with „Japan", „India", „Korea" and „China". When you click on any of these buttons, the text present on it will get displayed in the text field with „Times", „Courier", „Arial", „Verdana" fonts respectively. Use any shade of red as the foreground color.
18. Display four different text boxes on the applet. First is to input the name of font, second is to input foreground color, third to input background color and fourth to input the text. After pressing the „Enter" from the keyboard, the inputted text will be displayed at (150, 250) positions with respective combination of Colors and Font.
19. Display name of your college on the applet with „Lucida Bright" font, 20, UNDERLINED at 120, 200 positions. Set foreground color as Dark Green and background color as Yellow. After pressing „F10" key each time, the font of the text will get increased. Use „escape" key to exit from applet.
20. Display three concentric circles on the applet with different colors, excluding standard colors provided by compiler. Display the string „Circle" at the center of circles with font „Tahoma", 30, BOLD, UNDERLINED, ITALIC, GREEN. Use any shade of blue as the background color.

Question Bank:

Title: Program based on MenuBar, Menu, Checkable menu item and MenuItem

1. Create three Menus- 'Format', 'Tool' and 'Help' and add the suitable Menu items under these menus.
2. Create a menu oriented program which contains checkable Menu items- 'open' under the menu 'File' and ' Cut' menu item under the Menu 'Edit'. Add other items by your own.
3. Create a menu oriented program that simulates first three menus of a notepad application.
4. Develop a program to create three Menus- 'File', 'Edit' and 'Format'. Disable the 'Format' Menu.
5. Develop a program to create two menus-'File' and 'Edit'. Add the suitable menu items and the sub menu items. Make the use of checkable menu item.
6. Create a menu based program to display 'Favorites', 'Tools' and 'Help' menus of Windows explorer.
7. Create a menu based program to display 'Favorites', 'Tools' and 'Help' menus of Windows Internet explorer.
8. Create a menu based program to display 'Favorites', 'Insert' menus of Microsoft Office Excel.
9. Create a menu based program to simulate 'play' menu of Windows media player.
10. Create a menu based program to show 'view' and 'insert' menu of Microsoft Office Access.
11. Create a menu based program to simulate all menus of Windows calculator.
12. Create a menu based program to display 'view', 'image' and 'colors' menus of Microsoft Paint.
13. Create a menu based program to show 'file' menu of Outlook Express.
14. Create a menu based program to show 'format' and 'tools' menus of Microsoft Office PowerPoint.
15. Create a menu based program to simulate 'view' and 'clip' menus of Windows Movie Maker.
16. Create a menu based program to show all the menus of Windows Messenger.
17. Create a menu based program to show 'arrange' and 'Table' menus of Microsoft Office Publisher.
18. Create a menu based program to show all the menus of 3D pinball for windows game.
19. Create a menu based program to show 'view' and 'picture' menus of Microsoft Office Picture Manager.
20. Create a menu based program to simulate menus of Windows Minesweeper game.

Question Bank:
Title: Program based on Checkbox, CheckboxGroup, List, Choice and Scrollbar

1. Create four checkboxes named „INDIA“, „CHINA“, „KOREA“, „JAPAN“ and a text field with flow layout left. When you click, any of these checkboxes, the label of the respective check box will get displayed in the text field.
2. Display four different checkboxes with the names of different colors. When you click any of these checkboxes, the background color of the applet will get changed. Only one item can be selected at a time.
3. Display four checkboxes and a choice control on the applet. After clicking on the respective checkbox, the label associated with it will get copied in the choice control. After un-checking the checkbox, it will be removed from the choice. Each time the status bar will display the action string in it.
4. Display a list and a choice control on applet. Display two buttons in between them named „Remove“ and „Add“. After pressing the buttons, the top item of the list will get inserted into choice deleting from it and vice versa.
5. Display two scroll bars, a rectangle and a circle on the applet. Use the values of the scrollbars to set the size of the rectangle and circle respectively.
6. Display three scrollbars on applet. These will control the RGB values of background color. Use them to set the color of applet background.
7. Display a text field, a button and a choice control on the applet with flow layout right. The button will contain label "–>". After pressing this button, the text entered in the text box will get inserted in the choice.
8. Display two text boxes, a check box, a button and a list on the applet. After pressing the button text box1“s string will be set to check box and text box2“s string will be inserted into the list. If any of the text fields is empty flash the error in the status window.
9. Display a text field, a button and a scrollbar on the applet. Use scrollbar to set the font size of the text inserted from text field. After pressing the button, the inserted text will get displayed on (100, 150) position with respect to current positions of the scrollbar as font size.
10. Apply the grid layout to four different check boxes with insets positions at (50, 40, 50, 40). The check boxes are named with „MAGENTA“, „CYAN“, „PINK“ and „GRAY“. When we click any of these check boxes, the background color of the applet will be changed with respective color name. We must take care that only one checkbox will be clicked at a time. Unless it will flash the error at status bar and won“t set the background color.
11. Display button on the applet with grid layout. When we click on it, a frame will be displayed with a scrollbar. After setting the maximum position to the scrollbar, the frame will be made invisible.
12. Create a frame with a list control, a button and a choice control. The list and choice both will contain names as „New“, „Exit“, „Open“ and „Save“. When we click on the button, the selected name of both list and choice must be „Exit“ to make the frame invisible.
13. Display a text field, a button, a list and a choice control on the applet. Set echo character of the text field as „$“. After pressing the button on the applet, the text will be inserted in either list or choice depending upon this

condition:- if first letter of the text is „vowel“ it will be inserted into list otherwise it will be inserted into choice control.

14. Create a list control with values as, “A”, “B”, “C”, “D” and “E”. Display a button and a scrollbar near to it. After selecting the name from the list, it will be displayed at position 100, 100 on the applet. Afterwards, use scrollbar to control the position of this name. The button is named with string „Reset“ which will reset the position of the name to 100, 100 again.

15. Create two choice controls which are filled with digits from 0 to 9 each. Display four buttons named „+“, „–“, „*“ and „/“. After pressing any of these buttons perform respective mathematical operation on the selected numbers from the choice controls and display their result in the text field.

16. Create a list control and two text fields on the applet. The list contains items such as, „Mumbai“, „Nagpur“, „Pune“, „Aurangabad“ and „Nashik“. When we click on the item name its will be displayed in first text field and when it is double clicked its name will be displayed in second text field.

17. Create two checkbox groups containing 10 items each which contains the numbers from 0 to 9 each. Display four buttons named „+“, „–“, „*“ and „/“. After clicking any of these buttons, the respective operations will be performed on the current selected radio button from the groups and the result will be displayed in the text area.

18. Display a choice control with the names of different fonts as „Arial“, „Courier“, „Verdana“ and „Times“ etc. A text is entered from the text field. After selecting the respective font, this text will get displayed in the same text field with that font.

19. Display a scrollbar, a choice and a text field on the applet with flow layout left. The choice control will contain 10 different items. Scrollbar will be used to make auto-selection from the choice control. That is, when scrollbar is moved, the next / previous item from the choice will be selected as per the displacement of the scrollbar. Display the current selected item in the text field.

20. Display a choice control, a list control and a radio button on the applet. These all controls will contain “50”, “100”, “150”, “200”, and “250” values each. We need to make selection of values from all the three controls. The background color of the applet will get changed from the RGB combination of these values.

Question Bank:
Title: Extending AWT Components

1. Display a text field on the applet. After making any selection from the checkbox displayed near to it, text will get displayed on status bar.
2. Create a scroll bar on applet. When we change its position the current thumb position of it will get displayed on the status bar.
3. Display a button on the applet. When you press button the total number of clicks will get displayed on the status bar.
4. Display four checkboxes on the applet. Create their group. When we click any of these check boxes, their label must get appeared on the status bar.
5. Enter the text in the text field and then press a button near to it. Then the text will get displayed in the text area.
6. You need to enter the text in the text field; the same will be displayed on the status bar after pressing button near to it and „=“ will be printed in the text box for each character.
7. Display the choice control on the applet. It will contain five different names. When you select any of these names, it will be displayed on the status bar.
8. Display a list control on the applet. After we make the selection by double clicking the item name must be displayed in front of the string “Your Selection is: ” on the applet.
9. Display the list control on the applet. It will contain five different names. When you select any of these names, it will be displayed on the status bar
10. Display a choice control on the applet. After we make the selection by clicking the item name, its index must be displayed in front of the string “Your Selection is: ” on the applet.
11. Create an event driven scroll bar to display its current position in the text box displayed near to it.
12. Display a text field on the applet. Accept a number from it. Assign this position to the scrollbar displayed near to it. After adjustment has made.
13. Enter the text in the text area. When you press the button near to it, the total number of digits that text contains, will get appeared in the status bar.
14. Display a choice control on the applet. It contains different names of colors. When we make the selection of it, the applet will display your selection as, “Your Selection color is: ”.
15. Create the radio buttons with different names. Display your current selection on the status bar.
16. Display a list on the applet containing six different items. When we make any selection from it, the selected name will be displayed on the applet with Courier New, 20, ITALIC.
17. Display a text field on the applet. We will enter the name in it. After pressing the button displayed, the text from the text field will appear as the label for the checkbox displayed near to it.
18. Display two text fields on the applet. Three checkboxes are also displayed with names „Add“, „Sub“ and „Mul“. When we make the selection from them. The related operation will be performed and display result on the status bar.
19. Create two different choice controls on the applet. They will contain different numbers; we need to make the selection from both. After second choice control“s selection, their addition will be appeared on the status bar.
20. Make the changes on the label of the checkbox as the position of the scroll bar is changed on the applet.

Question Bank:

Title: Socket Programming (Socket and ServerSocket)

1. Copy the contents of your current working file from server to client using port number 6001. The client will respond to it by sending the message: "Thank you". Use local host.
2. Establish a client server communication through port 5525. The communication will be completed after the server sends message: "END".
3. Send a request to the server from client to send its date. When the request is reached the server. The server will respond to the client.
4. Establish a client server communication with port number 1891. The client will send the size of your current working file to server and server will respond to it by doing the same.
5. Send a request to the server from client to send its date. When the request is reached the server. The server will respond to the client.
6. Read the sent message from the client. The server will read the message and encode it its format. It will resend the same message by encoding it. Use encoding procedure as for example: "INDIA" encoded as "JMEJB".
7. Use port number 13 for communication between client and server. Server will send its date and time to client after the request has received.
8. Establish a client-server communication to use port number 2041 as „analyze" port. That is, the client will send a string to local host on this port number. The server will analyze the string and respond by sending total number of upper case letters, lower case letters, digits and special symbols in it.
9. Create a communication between client and server. Client will send a request on port number 3642 for communication. Then server will ask for the password. The communication will not be established until the password is correct. After inputting correct password, the server will send: "Welcome" and close the communication.
10. Open the socket on sever and client sides on port number 4233. The server will send the data continuously to client until ctrl+c is pressed.
11. Create a echo server on port number 7. That is, when server receives any data the same data will be sent to the client.
12. Program a simple chatting application on the local host. The communication will be completed after the server sends message "QUIT".
13. Transfer a file from server to client. The client will save this file at path given by the server.
14. Open the server sockets on ports 5322 and 1204. Use the client application to check whether any server is available on these ports for communication or not.
15. Establish the communication of client and server on port 5005. When client requests the data on this port. Server will respond to it by sending its inetaddress information. Print this information on the client side.
16. Create a server and two clients on local host. Serve both the clients one by one by server by establishing the communication.

17. Read the sent message from the client. The server will read the message and encode it its format. It will resend the same message by encoding it. Use encoding procedure as for example: "INDIA" encoded as "JMEJB".
18. Use port number 13 for communication between client and server. Server will send its date and time to client after the request has received.
19. Establish a client server communication through port 5525. The communication will be completed after the server sends message: "END".
20. Establish a client-server communication to use port number 2041 as „analyze" port. That is, the client will send a string to local host on this port number. The server will analyze the string and respond by sending total number of upper case letters, lower case letters, digits and special symbols in it.

Prof.B.B. Jagadale
(Subject Teacher)