

Ahmedabad University
School of Engineering and Applied Science
BTech ICT Semester – II
CSC102 Object Oriented Programming
(Course Website - <https://sites.google.com/ahduni.edu.in/oop2019>)

Theory Assignment – 1

Last date of Submission: Tue, February 12, 2019

INSTRUCTIONS:

- Write your answers in A4 size assignment and organize the work in a file.
- Write 16 questions (Any 8 questions from Part-A and any 8 questions from Part-B). Attempt questions that you find most difficult. For Part-B questions, you can write their solutions on paper, or develop their programs in NetBeans and submit on Moodle.
- Answer all the questions briefly and to the point answers.
- Write sample code wherever possible/applicable.
- Copying assignment answers doesn't help you to learn the concepts. Hence, write answer in your own words. Any type of plagiarism/copy is not allowed.
- In case of any doubt, contact the course faculty – Prof. Aditya Patel

Assignment Topics: Introduction to Java and Object Oriented Programming, Classes and objects, Methods, Constructor, Destructor, Static variables, final variables, packages, access modifiers/visibility control, Polymorphism, Method overloading, Inheritance, Method overriding, Aggregation/Containership

PART-A – THEORY/CONCEPTUAL QUESTIONS

1. Explain in brief any 5 important features of Java language.
2. Explain important differences between C and Java (point-wise answer).
3. Draw the Java platform architecture figure including JDK, JRE and JVM. Write one line each on JDK, JRE and JVM (Refer the handout provided on course website).
4. How Java is a platform independent language?
5. Write the names, size (memory requirement) and range of values that can be stored in different primitive data-types in Java. Also, mention the default value of each data type. Why the size of character data type is 2 bytes in java?
6. Differentiate narrowing and widening conversion between data-types with example code. Why Java compiler gives error (possible lossy conversion), when you write the following statement:
float f = 3.14; (possible loss of precision)
7. What is Java API? What are the advantages/benefit of having API in programming and software development in general?
8. Explain the use of any 5 built packages part of Java API and their use.
OR
Explain any 5 packages in Java API library with their purpose.

9. What is System.in and System.out? Explain their usage.
OR
Explain the standard input stream and standard output stream in Java with their usage.
10. What are Wrapper classes or Object data-types and what is use of such classes in Java API/library?
Give sample code/snippet of a program where Wrapper class is used.
11. What are the drawbacks/limitations of procedural programming?
12. What is Object Oriented Programming and what are the advantages of designing and developing a program using OOP approach.
13. Give differences between procedure-oriented programming using C and Object Orientated Programming using Java.
14. Explain the OOP fundamental concepts of encapsulation and data hiding with example.
OR
What is a Class and how class provides encapsulation and data hiding. Take any good problem definition and write its solution using class, data-members, constructor and method definition. In main() method, create objects from the Class and invoking the class method using object/reference variable.
15. Give differences between Class and Object. Explain the syntax to instantiate/create an object from the class.
16. Explain the concept of data abstraction with example? Why class is also called as abstract data type?
17. Explain String class and its purpose. Explain any 5 methods with their prototype and usage.
18. Explain the use/purpose of defining a constructor in a class with sample code. What is a default constructor and what is a parameterized constructor. How the constructor is invoked/used in main() method?
19. Why Java does not have DELETE operator.
OR
What is automated memory management in Java and how it useful?
OR
Explain garbage collection in Java and the use of finalize method.
20. What is destructor and when you should define a destructor in class? Who invokes the destructor?
21. Explain the use of static variables and static methods and when they should be used? Explain with an example.
OR
What are the advantages of defining static variables and static methods?
22. Why main method must be declared as public and static?
23. Explain method overloading with example? When it should be used?
24. Why method overloading is known as compile time polymorphism?
25. Explain symbolic constant and whey they should be used. Also explain final method and final class.
OR
Explain the use of final keyword in Java. Explain (final variable, method and class)
26. Give differences between compile time polymorphism and runtime polymorphism. How can you achieve runtime polymorphism in Java program?
27. What are accessor and mutator methods in a class? Why they are defined?
OR
What are getter and setter methods in Java?

28. Explain the concept of package with sample code. What are the benefits of creating packages in an application?
29. Discuss the different levels of access protection in Java with sample code.
OR
Explain the use of public, private, protected and default access modifiers or visibility specifiers.
30. What is inheritance and what are its advantages? Explain different types of inheritance?
31. What are the drawbacks of multiple inheritance?
OR
Why multiple inheritance is not allowed in Java?
32. Explain the causes of the following exceptions with sample code: `ArrayIndexOutOfBoundsException`, `NullPointerException`, `NumberFormatException`
33. Object oriented thinking – Consider the simplified problem definition of online shopping system like flipkart or amazon. (you may assume/add additional requirements). Identify the classes and data-members from the following problem definition:

A customer views Products according to product category. One customer gives many orders to the company. Order can be cash on delivery order or credit order. Order contains order items with different quantities. After confirming Order, Payment can be made by COD or online banking or credit card. After payment is completed, Invoice is generated and stock of the item is reduced.

34. Object oriented thinking – Consider the simplified problem definition of information system for a College (you may assume/add additional requirements). Identify the classes and data-members from the following problem definition:

A College has many departments, each department runs many Programmes, Programme has many subject/courses. Each department has many faculty members. Each course is taught by one or more faculty. Students get enrolled courses (offered in a particular Semester) and get the course result at the end of Semester.

PART-B - PRACTICAL QUESTIONS

1. Write an OOP based program which accepts the data of 5 cricket players as follows. Apply the concept of array of objects.
- a) Name of the player
 - b) Total Runs scored by player
 - c) Number of innings played
- Calculate the batting averages of all the players and display them. Also display the details of the players having maximum batting average and minimum batting average.
2. Write OOP based program by defining the Class, data-members, constructors and methods using OOP concepts for the following problem definition.
- The program should allow to store distance in feet and inches as unit of measurement and allow to perform conversions from feet and inch.
- The program should also allow to store distance in meters and cms and provide conversions.
- Class Distance – Feet and inches.**

Class Length – meter and cm.

(1 feet = 12 inches, 1 meter = 100 cm, 1 meter = 3 feet, 1 inch = 0.39370 cm,

1 inch * 2.54 = 1 cm,

65 inches * 2.54 = 165 centimeters.

E.g.: If a baby is 64 centimeters long, what is her length in inches?

64 centimeters * 0.39 = 25 inches)

Develop methods to provide following functionality:

- Parameterized constructor taking feet and inch as argument
- Parameterized constructor taking another object of Distance as argument (copy constructor) – object reference as argument
- Allow the addition of new distance to the existence distance object using method overloading and static methods
 - void addFeet(int feet)
 - void addInches(int inches)
 - void add(int feet, int inches)
 - void add(Distance distObject)
 - static Distance add(Distance distObject1, Distance distObject2)
 - Convert basic int type to Distance class – void convertToDistance(int feet).
 - **Convert Distance class to Length class – Length convertToLength()**
 - Convert inches to feet and feet to inches
 - static int feetToInches(int feet)
 - static int inchesToFeet(int inches)
 - display() method to display the distance in following format: “Distance is 6 feet and 6 inches”
- Methods in in Length class:
 - void subtract(int meter, int cm)
 - void subtract(Length lenObject)
 - Convert basic int type to Length class – void convertToLength(int meter).
 - Convert Length class to basic int type.
 - **Convert Length class to Distance class – Distance convertToDistance()**

3. Write OOP based program by defining the Class, data-members, constructors and methods using OOP concepts for the following problem definition.

The program should provide matrix addition, multiplication and transpose of any n * n matrices like 2 * 2 as well as 3 * 3 matrices. Provide following constructor and methods in class Matrix:

1. public Matrix(int[][] data, int rows, int cols)
2. Matrix add(Matrix m2) – perform validation that both the matrices should have same rows and columns for addition
3. Matrix static add(Matrix m1, Matrix m2) – perform validation
4. Matrix multiply(Matrix m2) - perform validation that the number of columns of matrix1 must be same as number of rows of matrix2 to allow the multiplication
5. Matrix static multiply(Matrix m1, Matrix m2) – perform validation
6. void transpose() – replace the rows with columns of original matrix.

$$\begin{pmatrix} 5 & 4 & 3 \\ 4 & 0 & 4 \\ 7 & 10 & 3 \end{pmatrix}^T = \begin{pmatrix} 5 & 4 & 7 \\ 4 & 0 & 4 \\ 3 & 4 & 3 \end{pmatrix}$$

4. Write OOP based program using classes and objects for the following problem definition. Define a class named **ComplexNumber** which abstracts complex numbers. Complex numbers have a **real** and an **imaginary part**, both of which to be represented by int type in your class definition.

E.g.

5+10i

1+02i

6+12i

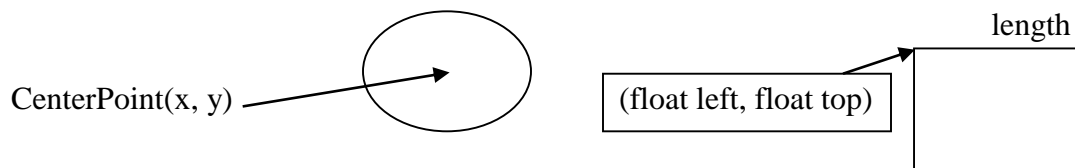
- Define **add** and **subtract methods** which both take an object of type ComplexNumber and return the *this* reference after performing the addition or subtraction.
 - **Also, define static add and subtract functions** that take two ComplexNumber objects and return a new object.
 - Additionally define the **equals** method, not to mention the toString() method to display the complex number object in the form a + ib.
 - Define at least three constructors; a default one, one taking two int parameters for the real and the imaginary parts, another one taking an object of ComplexNumber class as parameter.
 - Define a main class to test each of the methods of the ComplexNumber class and print out the results.
5. Write an OOP based program to conduct an online election contested by 5 candidates. The candidates are numbered 1 to 5 and the voting is done by marking the candidate number in front of candidate name on the ballot paper. Program should read the ballots and count the votes cast for each candidate and declare the winner based on maximum votes.
Hint: Define class Election with following data-members and methods:
- List of candidate codes as int array and list of candidate names as String array
 - List of votes casted for each candidate as int array.
 - Maximum Voter count in the area/constituency (as static variable)
 - void casteVote(int candidateCode)
 - void countVotes() – declare the winner and vote count for each candidate as well as total votes casted
 - void displayBallotMenu() – displays the names of candidates contesting the election and their codes (like 1,2,3,4,5) to be entered by the voter to caste vote against each candidate.
6. Define 2 classes **Circle** and **Square** to represent the shapes in 2D coordinate system. Circle class should have data-members like radius, x and y coordinators (in float) to represent the center of the circle. Circle should have methods like:
- constructor method (having arguments radius, x, y to initialize the data-members when object is created) – provide validation that x and y must between 1 and 800 pixels, otherwise default value of x=400 and y=400.

- method having signature `float findCircleArea(float r)` – use the formula $\text{PI} * r * r$ and uses the symbolic constant `PI` (3.14159) defined in the program. Define it as private static and final variable
- boolean `move(int x2, int y2)` – moves the center of circle to `x2, y2` -- provide validation that the `x2, y2` should be between 1 and 800 pixels (resolution of the screen) for valid move of the circle
- void `displayCircle(void)` method.

Define **Square** class having data-members like `length`, top-left point (`float left`, `float top`) to represent the left and top distance of the square from the center of the 2D axis.

- Define `findArea()` method – return `length * length`
- Also, define constructor method to initialize the data-members using parameterized constructor.

Create Circle or Square objects and display them (console based output, Non-GUI display) based on the inputs given by the user.



7. A book shop maintains the inventory of books that are being sold at the shop. The list includes details such as author, title, price, publisher and Quantity. Whenever a customer wants a book, the sales person inputs the title or author and the system searches the list and displays whether it is available or not. If it is not, an appropriate message is displayed. If it is, then the system displays the books details and requests for the number of copies required. If they are available, the total cost is displayed. Otherwise "requested copies not in stock" is displayed.

Design an OOP based program using a class called `BookStore` containing `main()` method and class `Book` (encapsulating book details) with suitable member functions and constructors.

8. Write OOP based program using classes and objects for the following problem definition. A team of 5 salesman are selling a product (e.g. RO Water Purifier) of a Company in a city. The team has a common stock of products (say 25 units of product) that they are supposed to sell out and achieve the target in a time period (like week). The program has to keep track of individual sale of each salesman as well as group sale. Whenever a sale is made, the stock (product qty) is reduced. Define following class and methods as below. Do not write code for the GUI. GUI is for explanation purpose only.

Define **Salesman** class with following data member:

`sno`, `sname`, `individualSale`, `groupSalesTarget`, `groupSale`, `itemStockQty`

`groupSalesTarget`, `groupSale`, `itemStockQty` are static-data member which will be shared and accessed by all Salesman objects.

Methods:

- a) Parameterized constructor with takes the information about Salesman and assign the initial values to data-members like sno and sname.
- b) generateSale(), that increase the total sale of the sales person by 1 and also increase the groupSale by 1.Reduce the itemStockQty by 1.
- c) Write one static member function that will check and display whether the group total sale achieve target or not and return true or false.
(boolean checkGroupSalesTarget())
- d) display() that will display all information about the individual salesman object as well as group information.

In the main method (), take an array of 5 salesman objects. And make the different salesman make sales, and display whether the target is achieved or not.

Salesman Program

Salesman Name: Harsh

Product quantity sold: 15

Stock Quantity: 85

Is Target Achieved: False

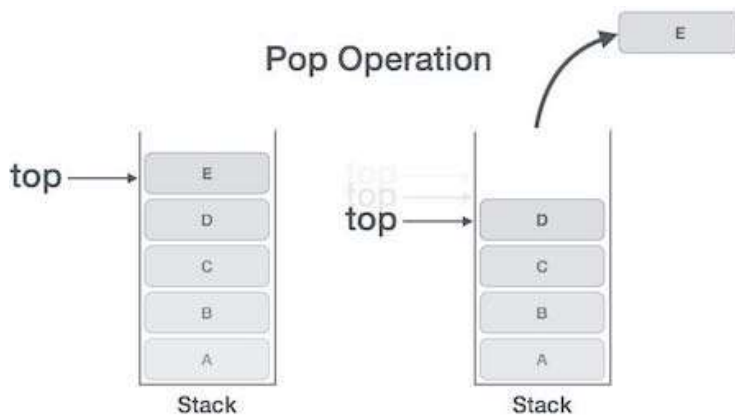
Total Qty Sold = 15

Target = 20

Set Target

Generate Sale

9. Write an OOP based program to implement a data structure called Stack (real-life examples of Stack are Email Inbox/SMS inbox, pile of dinner plates in the kitchen. Stack is a container/aggregate class in which user can store many elements and provides LIFO (Last-in First-out) of elements i.e. the last element added is the first to be removed/popped. Provide following operations in Stack class - 'push' operation is used to add an element to top of stack and 'pop' operation is used to remove and return the top most element from stack. 'Peek' operation is also implemented returning the value of the top element without removing it. Create a **stack of integers**.



10. Design appropriate class and methods in order to store a polynomial and perform the following operations:

- Add two polynomials
- Subtract two polynomials
- Multiply a polynomial two polynomials
- Multiply a polynomial with a constant

11. Write an OOP based program to store items of Reliance Shopping mall with details item no, name and price. Provide the functionality of inserting, deleting and updating the items. Also provide the functionality of searching a product with item_no, name and price.

Hint: Define Product class and use ArrayList to manage multiple items in the shopping Mall.

12. A bank provides 2 kinds of accounts for customers, one called as savings account and the other as current account. Both the accounts provide withdraw and deposit functionality.

The savings account provides interest rate (3% is fixed) and limited number of transactions (i.e. deposit and withdrawals) in a month (10 per month).

The current account provides an overdraft facility i.e. customer can withdraw up to certain overdraft certain (INR 50,000), even if the balance is 0. Current account does not provide any interest rate and provides unlimited number of transactions. Customer can print their passbook i.e. the list of transactions done in a bank account.

Open one account of current type and one account of saving type with sample data, perform banking transactions, print passbook/transaction log and display their information on the console.

13. Vehicle Parking Digitization

Write a program to help gate-keeper so that he can keep track of all the different vehicles parked in the parking-area and their different parking charges (based on type of vehicle and parking hours). Vehicles can of different types like 2-wheeler, 3-wheeler, 4-wheeler, each will have different parking charges.

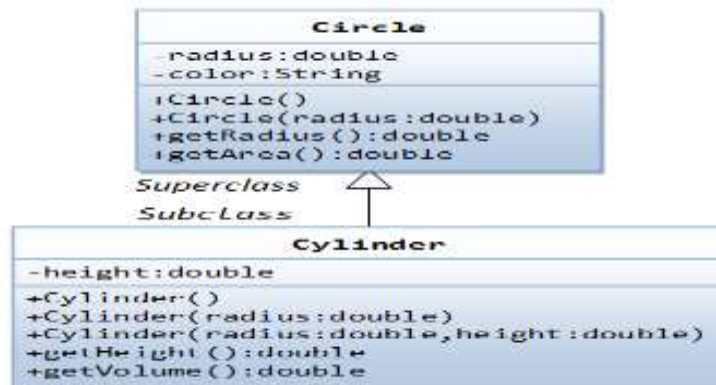
When new vehicle enters, the count of that category of vehicles is incremented. Your program should fulfill following queries:

- Category-wise count of number of vehicles, that are currently in parking area

- To check whether a particular Vehicle No.(GJ-1 2345) is the parking lot or not.
- When a vehicle leaves, the parking charge should be generated based on number of hours parked.

Design your program so that it can handle 20 vehicles.

14. Inheritance and method overriding



Write a class **Circle** having data-members - **radius** static final double variable PHI (22.0/7).

Write parameterized constructor of Circle having radius argument.

Define following methods in Circle:

```

double area( ): PHI * radius * radius
double circumference( ) : 2 * PHI * radius
  
```

Define a sub-class **Cylinder** inheriting from **Circle** having data-member height.

Write parameterized constructor of Cylinder having arguments radius and height. Call parent class constructor from this constructor using **super(radius)**.

Define following methods in Cylinder:

```

double volume( ): PHI * radius * radius * height
  
```

over-ridden method:

```

double area( ) : circumference( ) * height;    //over-rides the inherited method area( )
of circle.
  
```

In main() method, create objects of Circle and Cylinder class with suitable data and invoke the methods in the classes.