

## Assignment 1

### Introduction to JAVA

**CO1:** Understand and implement fundamental and OOP features through Java and Python Programming

1. Write a program to accept two short integers from user and display the sum. Check what happens when the sum exceeds the maximum range of short.
2. Write a program that accepts radius of a circle and displays area of the circle. Overload the constructor to accept radius as input and another circle object as input. Then show the effect of shallow vs deep copy of objects. Declare a constant pi equals to 3.14 using OOP concept.
3. Input n and consider an array of 1 to n natural numbers. Skip every second value and print the resulting series. Then select every third value from the remaining numbers. Print the resulting series. Repeat this process till the skip count becomes greater than the length of the list.

{1,2,3,4,5,6,7,8,9,10}→{1,3,5,7,9}→{1,7}

4. Write a program that accepts a **String** and assigns it to another. Check the outcome of comparison with == and equals() method. Take two Strings and put same input for them. Repeat the equality checking. Observe the outcome.
5. Write a program where class contains **void show(int)** to display the argument passed. Call the function once with **short** as actual parameter and again **double** as actual parameter. Add another function as **void show(double)** . Repeat the calls. Observe the outcomes in each case.
6. Create a program for ordering Pizza. The user should mention the size of the Pizza and the toppings he/she wants. A user may ask for any toppings. Implement this using (i) variable arguments concept and (ii) command line arguments.
7. Design a **BankAcct** class with account number, balance and interest rate as attribute. Interest rate is same for all account. Support must be there to initialize, change and display the interest rate. Also supports are to be there to return balance and calculate interest.
8. For a programme (such as, BCSE), each Instructor has name and phone number. Each textbook has a title, author name and publisher. Each course (that is, subject) has a course name, instructor and text book.
  - One can set the data for a textbook and view the same.
  - One can view instructor information and set the information.
  - One can set the course data and view the same.

Design and implement the classes.