# Second Semester Theory Solutions

1. Design a class **Perfect** to check if a number is Perfect or not. [A number is said to be perfect if the sum of the factors of the number excluding itself is equal to the original number]

   **Class Name :** Perfect

   **Data Member**

   - **num :** To store the number

   **Member functions**

   - **Perfect(int nn) :** a constructor to initialize the data member num=nn.
   - **int sum_of_factors(int i) :** Returns the sum of the factors of the number num (excluding itself) using recursive technique
   - **void check() :** Checks whether the given number is perfect or not by invoking the function sum_of_factors() and displays appropriate message

   Specify the class **Perfect** giving all the details as specified. Also define the main function to create an object and call accordingly to enable the task.

```
import java.util.*;
class Perfect
{
    int num;

    Perfect(int nn)
    {
        num = nn;
    }

    int sum_of_factors(int i)
    {
        int n = num;
        if(i > n/2)
        {
            return 1;
        }
        else
        {
            if(n%i == 0)
            {
                return i + sum_of_factors(i+1);
            }
            else
            {
                return sum_of_factors(i+1);
            }
        }
    }

    void check()
    {
        int s=sum_of_factors(2);
        if(s==num)
        {
            System.out.println(num + " is a perfect number.");
        }
        else
        {
            System.out.println(num + " is not a perfect number.");
        }
    }
    public static void main(String Args[])
    {
        Scanner input = new Scanner(System.in);
        System.out.print("Enter a number : ");
        int n = input.nextInt();
        Perfect object = new Perfect(n);
        object.check();
    }
}
```

2. The sum of the angles is calculated as:

   Let the first angle 20 degrees and 45 minutes

   Let second angle = 12 degrees 40 minutes.

   The sum of angles will be 33 degrees 25 minutes, (where 60 minutes = 1 degrees.)

   class Angle is designed as :

   **Class Name :** Angle

   **Data member**

   - **deg, min :** integer variables to store degrees and minutes.

   **Member functions/ Methods**

   - **Angle () :** a constructor to initialize deg and min.
   - **void acceptAngle() :** to input deg and min using keyboard.
   - **void dispAngle() :** to print deg and min.
   - **Angle sumOfAngle(Angle T1, Angle T2) :** to find the sum of angles from current objects T1, T2 by using above methods of adding angles and return sum of angles.

   Specify the class **Angle** giving all the details as specified. Also define the main function to create an object and call accordingly to enable the task.

```
import java.util.*;
class Angle
{
    int deg;
    int min;

    Angle()
    {
        deg = 0;
        min = 0;
    }

    void acceptAngle()
    {
        Scanner input = new Scanner(System.in);
        System.out.print("Degree : ");
        deg = input.nextInt();
        System.out.print("Minute : ");
        min = input.nextInt();
    }

    void dispAngle()
    {
        System.out.println("Degree : "+deg+ " \nMinutes : "+min);
    }

    Angle sumOfAngle(Angle T1, Angle T2)
    {
        Angle sum = new Angle();
        sum.deg = T1.deg + T2.deg;
        sum.min = T1.min + T2.min;
        if( sum.min >= 60)
        {
            sum.min = sum.min - 60;
            sum.deg = sum.deg + 1;
        }
        return sum;
    }

    public static void main(String Args[])
    {
        Angle T1 = new Angle();
        Angle T2 = new Angle();
        Angle call = new Angle();
        System.out.println("Enter the first angle.");
        T1.acceptAngle();
        System.out.println("Enter the second angle.");
        T2.acceptAngle();
        System.out.println("The first angle : ");
        T1.dispAngle();
        System.out.println("The second angle : ");
        T2.dispAngle();
        call = call.sumOfAngle(T1, T2);
        System.out.println("The sum of the angles : ");
        call.dispAngle();
    }
}
```

3. A class Capital has been defined to check whether a sentence has words beginning with a Capital Letter or not. Some members of the class are given below:

   **Class Name :** Capital

   **Data Member**

   - **sent :** To store a sentence
   - **freq :** Stores the frequency of words beginning with a capital letter

   **Member functions/ Methods**

   - **Capital () :** Default constructor
   - **void input() :** to input a sentence
   - **boolean isCap(String W) :** Checks and returns true if the word begins with a capital letter otherwise returns false
   - **void display() :** Displays the sentence along with frequency of the words beginning with a Capital Letter.

   Specify the class **Capital** giving all the details as specified. Also define the main function to create an object and call accordingly to enable the task.

```
import java.util.*;
class Capital
{
    String sent;
    int freq;

    Capital()
    {
        sent = "";
        freq = 0;
    }

    void input()
    {
        Scanner input = new Scanner(System.in);
        System.out.print("Enter a sentence : ");
        sent = input.nextLine();

    }

    boolean isCap(String W)
    {
        char c = W.charAt(0);
        if(Character.isUpperCase(c))
        {
            freq++;
            return true;
        }
        else
        {
            return false;
        }
    }

    void display()
    {
        System.out.println(sent);
        System.out.println(freq);
    }

    public static void main(String Args[])
    {
        Capital object = new Capital();
        object.input();
        String words[] = object.sent.split(" ");
        for (int i = 0; i < words.length; i++)
        {
            object.isCap(words[i]);
        }
        object.display();
    }
}
```

4. Register is an entity which can hold a maximum of 100 names. The register enables the user to add and remove names from the topmost end only. k

   The details of the class **Register** is given below:

   **Class Name :** Register

   **Data members/ Instance variables**

   - **stud[] :** array to hold names of students
   - **cap :** maximum capacity of the Storage
   - **top :** to point the index of the top end

   **Methods/ Member functions**

   - **Register (int max) :** constructor to initialize cap=max, top=-1 and create the String array
   - **void push (String n) :** to add names in the register at the top location if possible otherwise display the message "OVERFLOW"
   - **String pop() :** removes and returns the name from the topmost location of the register if any else returns "$$"
   - **void display() :** displays the names in the register

   Specify the class **Register** giving details of the functions **void push(String)** and **String pop()**. Assume that the other functions have been defined.

```
import java.util.*;
class Register
{
    String stud[];
    int cap;
    int top;

    Register(int max)
    {
        cap = max;
        top = -1;
        stud = new String[cap];
    }

    void push(String n)
    {
        if(top == cap-1)
        {
            System.out.println("OVERFLOW");
        }
        else
        {
            stud[++top] = n;
        }
    }

    String pop()
    {
        if(top == -1)
        {
            return "$$";
        }
        else
        {
            return stud[top--];
        }
    }
}
```

5. A super class **Number** has been defined to calculate the factorial of a number. Define a subclass to find the sum of the series.

   The details of the members of both the classes are given below:

**Class Name :** Number

**Data members/ Instance variables**

- **n :** to store an integer number

**Methods/ Member functions**

- **Number(int nn) :** parameterized constructor to assign values to the data members
- **int factorial(int a) :** Returns the factorial of a number (factorial of n=1x2x3x4...xn)
- **void display() :** to display the data members

**Class Name :** Series

**Data members/ Instance variables**

- **sum :** to store the sum of the series

**Methods/ Member functions**

- **Series(...) :** parameterized constructor to assign values to the data members of both the classes
- **void calsum() :** to calculate the sum of the given series
- **void display() :** to display the data members of both classes

Assume that the super class **Number** has been defined. Using the concept of inheritance specify the class **Series** giving the details of the **constructor(...)**, **void calsum()** and **void display()**
The super class, main function and algorithm need NOT be written.

```
import java.util.*;
class Series extends Number
{
    int sum;

    Series()
    {
        super(nn);
        sum = 0;
    }

    void calsum()
    {
        for (int i = 1; i <= n; i++) {
            int fact = factorial(i);
            sum += fact;
        }
    }

    void display()
    {
        super.display();
        System.out.println("Sum of the series: " + sum);
    }
}
```