Interview Preparation Programs 01

1)SumOfDigits.java

```
package programs;

public class SumOfDigits {
    public static void main(String[] args) {
        //System.out.println(sumOfDigits(1234));
        System.out.println(sumOfDigits(-125));
        //System.out.println(sumOfDigits(19));
    }
    public static int sumOfDigits(int num){
        if(num <0){
            return -1;
        }
        int sum = 0;

        while(num >0){
            sum += num%10;
            //System.out.println("Num = " + num + ", num / 10 = " +
((float)num/10) );
            System.out.println("Num = " + num);
            num /= 10;
        }
        return sum;
    }
}
```

2)AreaCalculator.java

```
package programs;

public class AreaCalculator {
    public static void main(String[] args) {
        System.out.println(area(5));
    }
    public static double area(double radius){
        if(radius <0){
            return -1.0;
        }
        return 3.14159 *radius*radius;
    }
}
```

3)BytesToMegaBytes.java

```
package programs;

public class BytesToMegaBytes {
    public static void main(String[] args) {
        printMegaBytesAndKiloBytes(2000);
    }
    public static void printMegaBytesAndKiloBytes(int kiloBytes){
```

```
        if(kiloBytes <0){
            System.out.println("Invalid Value");
            return;
        }
        int megabytes = kiloBytes/1024;
        int remainingBytes = kiloBytes% 1024;
        System.out.println(kiloBytes + " KB = " + megabytes + " MB and " +
remainingBytes + " KB");
    }

    public static class LoopsTest {
        public static void main(String[] args) {
            forLoopTest();
            printEvenNumbers();
        }
        //Test without any init, expression, increment
        public static void forLoopTest(){
            int i = 1;
            for(;;){
                System.out.println("i = " + i);
                break;
            }
        }
        public static boolean isEvenNumber(int num){
            return num >0 && num%2==0 ? true:false;
        }
        public static void printEvenNumbers(){
            int num = 5;
            int evenCount = 0,oddCount = 0 ;
```

```
        while(num<=20){
            if(!isEvenNumber(num)){
                num++;
                oddCount++;
                continue;
            }
            evenCount++;
            System.out.println("num " + num + " is even");
            if(evenCount > 5){
                break;
            }
            num++;
        }
        System.out.println("Even count " + evenCount);
        System.out.println("Odd count " + oddCount);
    }

    public static class HelloWorld {
        public static void main(String[] args) {

        }
    }
  }
}
```

4)Factors.java

```
package programs;
```

```
public class Factors {
    public static void main(String[] args) {
        printFactors(217);
    }
    public static void printFactors(int number){
        if(number < 1){
            System.out.print("Invalid Value");
            return ;
        }
        for(int i=1;i<= number/2;i++){
            if(number%i==0){
                System.out.print(i + " ");
            }
        }
        System.out.print(number);
    }
}
```

5)FlourPacker.java

```
package programs;

public class FlourPacker {
    public static void main(String[] args) {
        System.out.println(canPack(1,0,5));
    }
    // write code here
    public static boolean canPack(int bigCount, int smallCount,
int goal){
```

```
        for(int i=0; i<= bigCount; i++){
            int bal = goal - i*5;
            if(bal ==0 || (bal >0 &&  bal <= smallCount)){
                return true;
            }
        }
        return false;
    }
}
```

6)GreatestCommonDivisor.java

```
package programs;

public class GreatestCommonDivisor {
    public static void main(String[] args) {
        System.out.println(getGreatestCommonDivisor(12,30));
    }
    // write code here
    public static int getGreatestCommonDivisor(int first, int second){
        if(first < 10 || second < 10){
            return -1;
        }
        int small = first <= second ? first : second;
        int big = first > second ? first : second;
        for(int i=small;i>=1;i--){
            if(small % i == 0 && big%i == 0){
                return i;
```

```
            }
        }
        return -1;
    }
}
```

7)InchesToCentimeter.java

```
package programs;

public class InchesToCentimeter {
    public static void main(String[] args) {
        int inches = 68;
        System.out.println(inches + " equals to = " +
inchesToCentiMeters(inches) + " cm");
    }

    public static double inchesToCentiMeters(int inches) {
        return 2.54 * inches;
    }
}
```

8)LeapYearAndDaysInMonth.java

```
package programs;

public class LeapYearAndDaysInMonth {
    public static void main(String[] args) {
        System.out.println("isLeapYear = " + isLeapYear(2024));
        System.out.println("getDaysInMonth = " +
```

```
getDaysInMonth(2,2024));
    }
    public static boolean isLeapYear(int year) {
        if (year < 1 || year > 9999) {
            return false;
        }
        if ((year % 4 == 0 && year % 100 != 0) || (year % 400 == 0)) {
            return true;
        }
        return false;
    }

    public static int getDaysInMonth(int month, int year) {
        if (month < 1 || month > 12 || year < 1 || year > 9999) {
            return -1;
        }

        return switch (month) {
            case 1, 3, 5, 7, 8, 10, 12 -> 31;
            case 4, 6, 9, 11 -> 30;
            case 2 -> {
                yield isLeapYear(year) ? 29 : 28;
            }
            default -> 0;

        };
    }
}
```

9)MilesPerHour.java

```
package programs;

public class MilesPerHour {
    public static void main(String[] args) {
        printConversion(1.5);
        printConversion(10.25);
        printConversion(-5.6);
        printConversion(25.42);
        printConversion(75.114);
    }
    public static long toMilesPerHour(double  kilometersPerHour){
        if(kilometersPerHour < 0){
            return -1;
        }
        return Math.round(kilometersPerHour/1.609);
    }
    public static void printConversion(double kilometersPerHour){

        if(kilometersPerHour < 0){
            System.out.println("Invalid Value");
            return;
        }
        long milesPerHour = toMilesPerHour(kilometersPerHour);
        System.out.println(kilometersPerHour + " km/h = " + milesPerHour
+ " mi/h");
    }
}
```

10)MinutesToYearsDaysCalculator.java

```
package programs;

public class MinutesToYearsDaysCalculator {
    public static void main(String[] args) {
        printYearsAndDays(100000);
    }
    public static void printYearsAndDays(long minutes){
        if(minutes <0){
            System.out.println("Invalid Value");
            return;
        }

        long days = minutes / (60 * 24);
        long years = days/365;
        long daysRemaining = days %365;
        System.out.println(minutes + " min = " + years + " y and " +
daysRemaining + " d");
    }
}
```

11)NumberToWords.java

```
package programs;

public class NumberToWords {
    // write code here
```

```
public static void main(String[] args) {
    numberToWords(100);
}
public static int getDigitCount(int number){
    if(number<0){
        return -1;
    }
    if(number == 0){
        return 1;
    }
    int count = 0;
    while(number !=0){
        count ++;
        number /=10;
    }
    return count;
}
public static int reverse(int number){
    int orgNumber = number;
    int sum = 0;
    while(number !=0){
        sum = sum*10  + number%10;
        number /=10;
    }
    return sum;
}
public static void numberToWords(int number){
    if(number < 0){
        System.out.print("Invalid Value");
        return ;
```

```
    }
    if(number == 0){
        System.out.print("Zero");
        return ;
    }
    int count =  getDigitCount(number);
    int reverse = reverse(number);
    int revCount =  getDigitCount(reverse);
    while(reverse !=0){
        int digit = reverse %10;
        String word = switch(digit){
            case 0 -> "Zero";
            case 1 -> "One";
            case 2 -> "Two";
            case 3 -> "Three";
            case 4 -> "Four";
            case 5 -> "Five";
            case 6 -> "Six";
            case 7 -> "Seven";
            case 8 -> "Eight";
            case 9 -> "Nine";
            default->"Invalid Value";
        };
        System.out.println(word);
        reverse /=10;
    }
    while(revCount < count){
        System.out.println("Zero");
        revCount++;
    }
```

```
    }
}
```

12)Palindrome.java

```
package programs;

public class Palindrome {
    public static void main(String[] args) {
        System.out.println(isPalindrome(-121));
        System.out.println(isPalindrome(121));
    }
    public static boolean isPalindrome(int number){
        int orgNumber = number;
        int sum = 0;
        while(number !=0){
            sum = sum*10  + number%10;
            number /=10;
        }
        return sum == orgNumber;
    }
}
```

13)PerfectNumber.java

```
package programs;

//A number whose sum of factors (excluding the number itself)
// 6 is a perfect number = 1+ 2+3
//28 is a perfect number = 1+ 2+4+7+14+28
//496 is a perfect number
//8128 is a perfect number


public class PerfectNumber {
    // write code here
    public static void main(String[] args) {
        System.out.println(isPerfectNumber(6));
        System.out.println(isPerfectNumber(5));
        System.out.println(isPerfectNumber(28));
    }
    public static boolean isPerfectNumber(int number){
        if(number < 1){
            return false;
        }
        int sum = 1;
        for(int i=2;i<=number/2;i++){
            if(number%i==0){
                sum +=i;
```

```
            }
        }
        return sum == number;
    }
}
```

14)PrimeNumber.java

```
package programs;

public class PrimeNumber {
    public static void main(String[] args) {
        int num = 10;
        System.out.println("Number " + num + " is Prime ? " +
isPrime(num));
        printPrimeNumbers();
    }

    //No, 1 is not a prime number. The number 1 has only 1 factor.
    // For a number to be classified as a prime number, it should have
exactly two factors.
    // Since 1 has less than two factors, it is not a prime number.
    public static boolean isPrime(int wholeNumber) {
        for(int i = 2; i <= wholeNumber/2;i++){
            if(wholeNumber%i == 0){
                return false;
            }
        }
        return true;
```

```
}
public static void printPrimeNumbers(){
    int count = 0;
    for(int i=2;count <3;i++){
        if(isPrime(i)){
            count +=1;
            System.out.println(count+ " Prime Number = " + i);
        }
    }
}
//Program to find to the largest prime factor of a number
public static int getLargestPrime(int number){
    if(number < 0){
        return -1;
    }
    for(int i = number;i>1;i--){
        if(number%i==0){
            boolean prime = true;
            for(int j = i/2;j>1;j--){
                if(i%j==0){
                    prime = false;
                }
            }
            if(prime){
                return i;
            }
        }
    }
    return -1;
}
```

```
}
```

15)ReverseArray.java

```
package programs;

import java.util.Arrays;

public class ReverseArray {
    public static void main(String[] args) {
        reverse(new int[]{5,4,3,2,1});
    }
    // write code here
    private static void reverse(int []b){

        System.out.print("Array = " + Arrays.toString(b));
        if (b.length <= 1) {
            return ;
        }
        for (int i = 0; i < (b.length / 2); i++) {
            int temp = b[i];
            b[i] = b[b.length - i - 1];
            b[b.length - 1 - i] = temp;
        }
        System.out.println("Reversed Array = " + Arrays.toString(b));

    }
}
```

16)SecondsToHoursAndMinutes.java

```
package programs;

public class SecondsToHoursAndMinutes {
    public static void main(String[] args) {
        System.out.println(secondsToMinsStr(-3945));
        System.out.println(secondsToMinsStr(-65,45));
        System.out.println(secondsToMinsStr(65,145));
        System.out.println(secondsToMinsStr(65,45));
        System.out.println(secondsToMinsStr(3945));
    }
    public static String secondsToMinsStr(int seconds){

        if(seconds <0){
            return "Invalid data for seconds(" + seconds + "), must be a
positive integer value";
        }
        return secondsToMinsStr(seconds / 60,seconds % 60);
    }
    public static String secondsToMinsStr(int minutes, int seconds){

        if(minutes <0){
            return "Invalid data for minutes(" + minutes + "), must be a
positive integer value";
        }
        if(seconds <=0 || seconds >=59){
            return "Invalid data for seconds(" + seconds + ")";
        }
```

```
        int hours = minutes/60;

        int remainingMins = minutes % 60;

        return hours + "h " + remainingMins + "m " + seconds + "s";
    }
}
```

17)SharedDigit.java

```
package programs;

public class SharedDigit {
    public static void main(String[] args) {
        System.out.println(hasSharedDigit(12,423));
    }
    public static boolean hasSharedDigit(int number1, int number2){

        /*if(number1 <10 || number1 >99 || number2 < 10 || number2 >99){
            return false;
        }*/

        int orgNumber2 = number2;
        while(number1 !=0){
            int a = number1%10;

            number2 = orgNumber2;
            while(number2!=0){
                int b = number2%10;
```

```
                if (a == b){
                    return true;
                }
                number2 /=10;
            }
            number1 /=10;
        }
        return false;

    }
}
```

18)SortedArray.java

```
package programs;

import java.util.Scanner;
import java.util.Arrays;

public class SortedArray {
    public static void main(String[] args) {

        int[] a = getIntegers(5);
        printArray(a);
        System.out.println(Arrays.toString(sortIntegers(a)));//descending
    }

    // write code here
    public static int[] getIntegers(int size) {
        Scanner scanner = new Scanner(System.in);
```

```
    int[] a = new int[size];
    System.out.println("Enter the numbers for array size " + size);
    for (int i = 0; i < size; i++) {
        a[i] = scanner.nextInt();
    }
    return a;
}

public static void printArray(int[] a) {
    for (int i = 0; i < a.length; i++) {
        System.out.println("Element " + i + " contents " + a[i]);
    }
}

//Assumption - input array is sorted
//Output - Sorted numbers in descending order
public static int[] sortIntegers(int[] a) {

    int[] b = Arrays.copyOf(a, a.length);
    Arrays.sort(b);
    if (b.length <= 1) {
        return b;
    }
    for (int i = 0; i < b.length / 2; i++) {
        int temp = b[i];
        b[i] = b[b.length - i - 1];
        b[b.length - 1 - i] = temp;
    }
    return b;
}
```

```
    private static int findMin(int[] a) {
        int min = a[0];
        for (int i = 0; i < a.length; i++) {
            if (a[i] < min) {
                min = a[i];
            }
        }
        return min;
    }

}
```

19)Armstrong.java

```
Input:153
Output: Yes
153 is an Armstrong number.
1*1*1 + 5*5*5 + 3*3*3 = 153

371 is an Armstrong number
3*3*3 + 7*7*7 + 1*1*1 = 371

1634 is an Armstrong number
1*1*1 + 6*6*6 + 3*3*3 + 4*4*4 = 1634
```

```
package programs;
public class Armstrong {

    public static void main(String[] args) {

        int number = 371, originalNumber, remainder, result = 0;

        originalNumber = number;

        while (originalNumber != 0)
        {
            remainder = originalNumber % 10;
            result += Math.pow(remainder, 3);
            originalNumber /= 10;
        }

        if(result == number)
            System.out.println(number + " is an Armstrong number.");
        else
            System.out.println(number + " is not an Armstrong number.");
    }
}
```