Practical 3

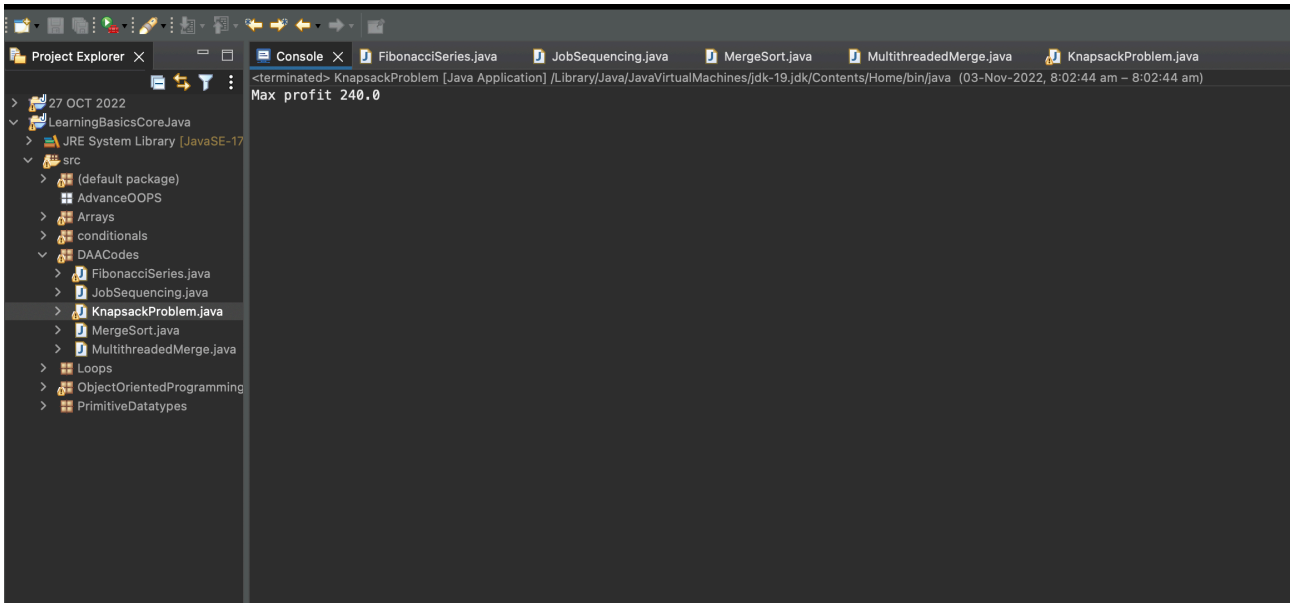
Practical - Write a program to solve a fractional Knapsack problem using a greedy method.

Code -

```
package sanketscode;

import java.math.BigDecimal;

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

		//1. Create Input values
		int m = 50; //capacity

		int[] Pi = { 60 , 100, 120};
		int[] wi = {10 ,20, 30};
		int n = wi.length;
		double[] piwi = new double[n];
		double ans = 0;
		double containedProfit = 0;


		for(int i=0;i < n;i++) {
			piwi[i] = Pi[i]/wi[i];
		}

		double bigPiwi = 0;


		do {
			bigPiwi = 0;
			for (int i=0;i < piwi.length;i++) {
				if(piwi[i] > bigPiwi) {
					bigPiwi = piwi[i];
				}
			}


			for(int i=0; i < n;i++) {
				if(bigPiwi == piwi[i]) {
					if((m - wi[i]) >= 0) {

						m = m - wi[i];
```

```
                    containedProfit+=Pi[i];

                }else {


                    containedProfit += Pi[i]*m/wi[i];
                    bigPiwi = 0;

                }
                 piwi[i] = 0;
            }
        }

    }while(bigPiwi != 0);

    System.out.println("Max profit " + containedProfit);



    }
}
```

Output -



Conclusion - These Problems can be solved using fractional knapsack problem.