**Activity-1**

**Consider the following program segment**

```
//import classes
public class Activity1
{
        public static void main(String [] args)
        {
                //variable declaration
                //executable statements
        }
}
```

a. Write Java statements that declare the following variables: **num1** , **num2** , and **num3** , and **average** of type **int** .
b. Write Java statements that store **125** into **num1** , **28** into **num2** , and **-25** into **num3** .
c. Write a Java statement that stores the average of **num1** , **num 2** , and **num3** into **average**.
d. Write Java statements that output the values of **num1** , **num2** , **num3** , and **average** .
e. Compile and run your program

**Activity-2**

**Consider the following Java program in which the statements are in the incorrect order. Rearrange the statements so that it prompts the user to input the length and width of a rectangle and output the area and perimeter of the rectangle.**

```
public class Activity2
{
Scanner console = new Scanner(System.in);
import java.util.*;
{
public static void main(String[] args)
int width;
System.out.print("Enter the length: ");
width = console.nextInt();
System.out.println();
int length;
System.out.print("Enter the width: ");
length = console.nextInt();
System.out.println();
area = length * width;
System.out.println("Area = " + area);
System.out.println("Perimeter = " + perimeter);
perimeter = 2 * (length + width);
int area;
int perimeter;
}
}
```

**Activity-3**

Suppose you want to develop a program that changes a given amount of money into smaller monetary units. The program lets the user enter an amount as a double value representing a total in dollars and cents, and outputs a report listing the monetary equivalent in the maximum number of dollars, quarters, dimes, nickels, and pennies, in this order, to result in the minimum number of coins

**Here are the steps in developing the program:**

1. **Prompt the user to enter the amount as a decimal number, such as 11.56.**
2. **Convert the amount (e.g., 11.56) into cents (1156).**
3. **Divide the cents by 100 to find the number of dollars. Obtain the remaining cents using the cents remainder 100.**
4. **Divide the remaining cents by 25 to find the number of quarters. Obtain the remaining cents using the remaining cents remainder 25.**
5. **Divide the remaining cents by 10 to find the number of dimes. Obtain the remaining cents using the remaining cents remainder 10.**
6. **Divide the remaining cents by 5 to find the number of nickels. Obtain the remaining cents using the remaining cents remainder 5.**
7. **The remaining cents are the pennies.**
8. **Display the result.**

**You are required to implement the above steps 1-8 in JAVA language**

**Activity-4**

**N** students take **K** apples and distribute them among each other evenly. The remaining (the undivisible) part remains in the basket. How many apples will each single student get? How many apples will remain in the basket?

The program reads the numbers **N** and **K**. It should print the two answers for the questions above.

| Input: | Output: |
|---|---|
| 6 | 8 |
| 50 | 2 |

**Activity-5**

A school decided to replace the desks in three classrooms. Each desk sits two students. Given the number of students in each class, print the smallest possible number of desks that can be purchased.

The program should read three integers: the number of students in each of the three classes, **a, b** and **c** respectively.

| Input: | Output: |
| --- | --- |
| 17 | 28 |
| 19 | |
| 18 | |

**Activity-6**

Given the integer **N** - the number of minutes that is passed since midnight - how many hours and minutes are displayed on the 24h digital clock?

The program should print two numbers: the number of hours (between 0 and 23) and the number of minutes (between 0 and 59).

For example, if **N = 150**, then **150** minutes have passed since midnight - i.e. now is **2:30 am**. So the program should print **2 30**.

**Activity-7**

A milk carton can hold 3.78 liters of milk. Each morning, a dairy farm ships cartons of milk to a local grocery store. The cost of producing one liter of milk is $0.38, and the profit of each carton of milk is $0.27. Write a program that does the following:

a. **Prompts the user to enter the total amount of milk produced in the morning**
b. **Outputs the number of milk cartons needed to hold milk (Round your answer to the nearest integer.)**
c. **Outputs the cost of producing milk**
d. **Outputs the profit for producing milk**

**Activity-8**

You found an exciting summer job for five weeks. It pays $15.50 per hour. Suppose that the total tax you pay on your summer job income is 14%. After paying the taxes, you spend 10% of your net income to buy new clothes and other accessories for the next school year and 1% to buy school supplies. After buying clothes and school supplies, you use 25% of the remaining money to buy savings bonds. For each dollar you spend to buy savings bonds, your parents spend $0.50 to buy additional savings bonds for you. Write a program that prompts the user to enter the pay rate for an hour and the number of hours you worked each week. The program then outputs the following:

a. **Your income before and after taxes from your summer job**
b. **The money you spend on clothes and other accessories**
c. **The money you spend on school supplies**
d. **The money you spend to buy savings bonds**
e. **The money your parents spend to buy additional savings bonds for you**

**Activity-9**

A cricket game is to be held in a stadium and there are five seating categories available for the audience. Class A seats cost $20, Class B seats cost $15, Class C seats cost $10, and Class D seats cost $5. You should write a JAVA program that asks how many tickets for each class of seats were sold and finally display the income generated income corresponding to ticket sales.

**Activity-10**

Write a program that reads an integer between 0 and 1000 and adds all the digits in the integer. For example, if an integer is 932, the sum of all its digits is 14.

**Enter a number between 0 and 1000: 999**

**The sum of the digits is 27**