Mohammad Ali Jinnah University Karachi
Department of Computer Science

LAB MANUAL

CS1421: Object Oriented Programming
LAB 01

Instructors
Safiyah Batool
Samia Bashir

Lab 1
Introduction to Java and Procedural Programming Constructs

Activity time-boxing

Task no.	Activity name	Activity time
1	Introduction to OOP and Java	20 min
2	Environment setup	15 min
3	Lab discussion	40 min
4	Walkthrough tasks	30 min
5	Think it on! (Practice Tasks)	30 min
6	Evaluation Tasks	30 min

Table 1: Activity Time-boxing

Overview

Introduction to Java

Java Versions

There are many java versions that have been released. Current stable release of Java is Java SE 11.

1. JDK Alpha and Beta (1995)
2. JDK 1.0 (23rd Jan, 1996)
3. JDK 1.1 (19th Feb, 1997)
4. J2SE 1.2 (8th Dec, 1998)
5. J2SE 1.3 (8th May, 2000)
6. J2SE 1.4 (6th Feb, 2002)
7. J2SE 5.0 (30th Sep, 2004)
8. Java SE 6 (11th Dec, 2006)
9. Java SE 7 (28th July, 2011)
10. Java SE 8 (18th March, 2014)

11. Java SE 9 (21st September 2017)
12. Java SE 10 (20 March 2018)
13. Java SE 11 (September 25th, 2018)

Java Platforms

According to Oracle, there are four platforms of the Java programming language

- Java Platform, Standard Edition (Java SE)
- Java Platform, Enterprise Edition (Java EE)
- Java Platform, Micro Edition (Java ME)
- JavaFX: *JavaFX* is a software platform for creating and delivering desktop applications, as well as rich Internet applications (RIAs) that can run across a wide variety of devices

Java Basics

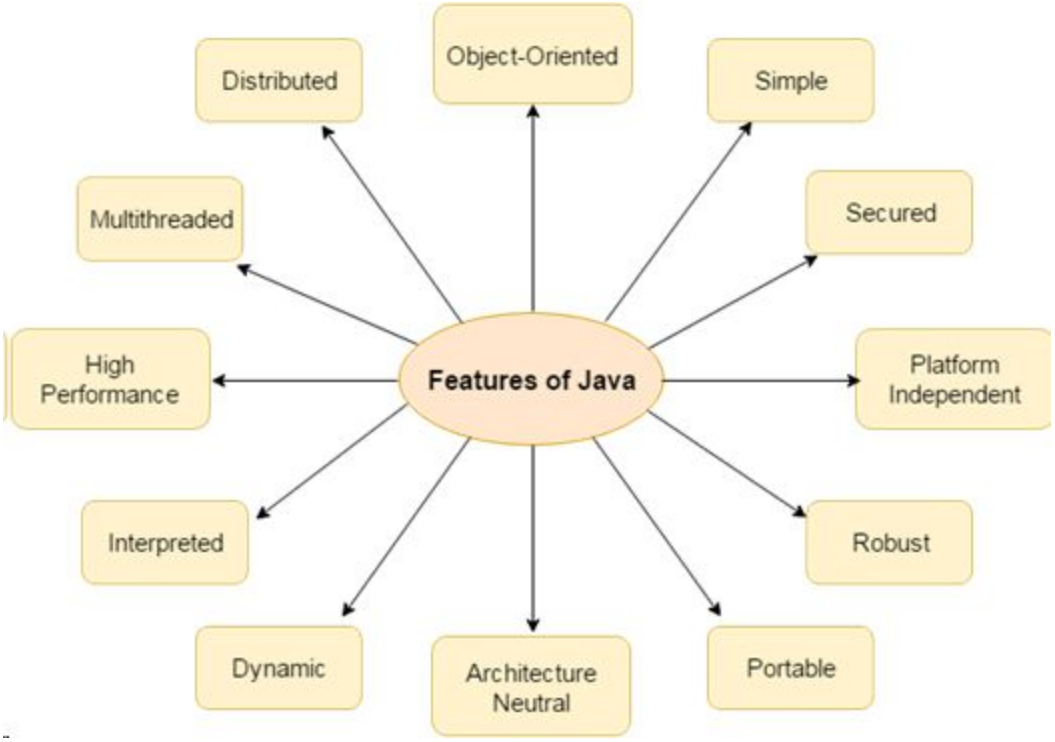
JDK, short for Java Development Kit, is a software development environment used for developing Java applications and applets. It includes the Java Runtime Environment (JRE), an interpreter/loader (Java), a compiler (javac), an archiver (jar), a documentation generator (Javadoc) and other tools needed in Java development.

JRE stands for **Java Runtime Environment.** The Java Runtime Environment provides the minimum requirements for executing a Java application; it consists of the *Java Virtual Machine (JVM), core classes,* and *supporting files.*

JVM A Java virtual machine (JVM), an implementation of the Java Virtual Machine Specification, interprets compiled Java binary code (called bytecode) for a computer's processor (or "hardware platform") so that it can perform a Java program's instructions. Java was designed to allow application programs to be built that could be run on any platform without having to be rewritten or recompiled by the programmer for each separate platform.

Just-in-time Compiler (JIT) is the part of the Java Virtual Machine (JVM) that is used to speed up the execution time. JIT interpret parts of the bytecode that have similar functionality at the same time, and hence reduces the amount of time needed for full interpretation.

Features of Java

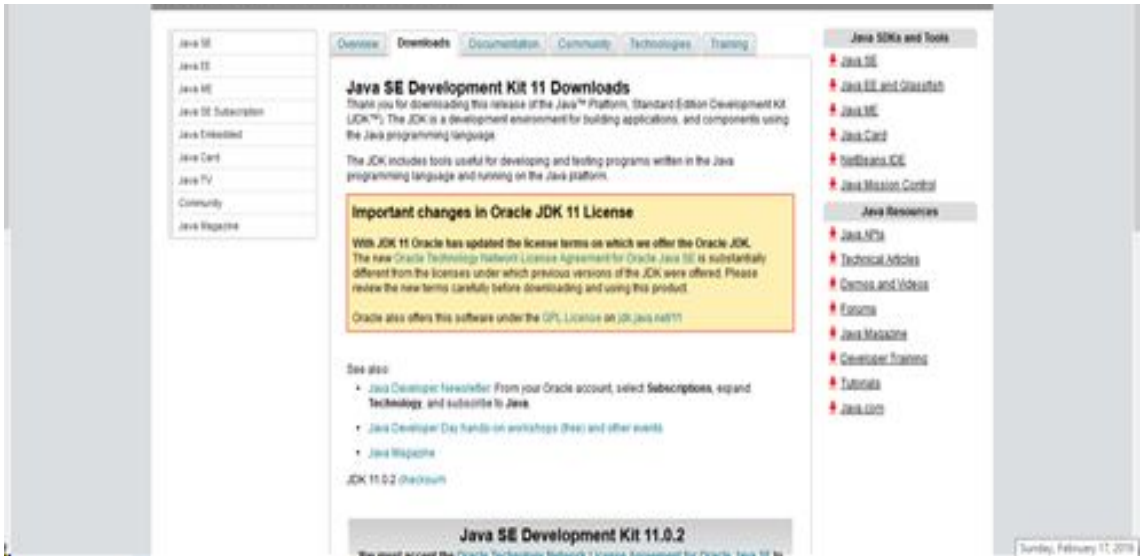


Environment Setup

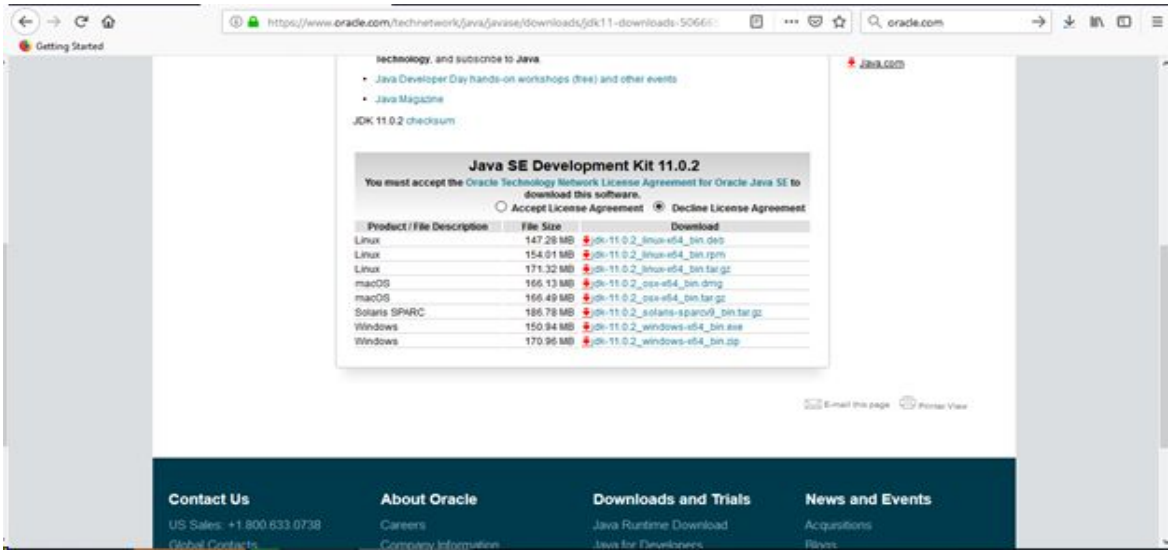
Java Development Kit

To develop Java applications on our computers, we require a JDK. Visit the link below to download the JDK setup.

Follow this link:
https://www.oracle.com/technetwork/java/javase/downloads/jdk11-downloads-5066655.html



Select Accept License Agreement

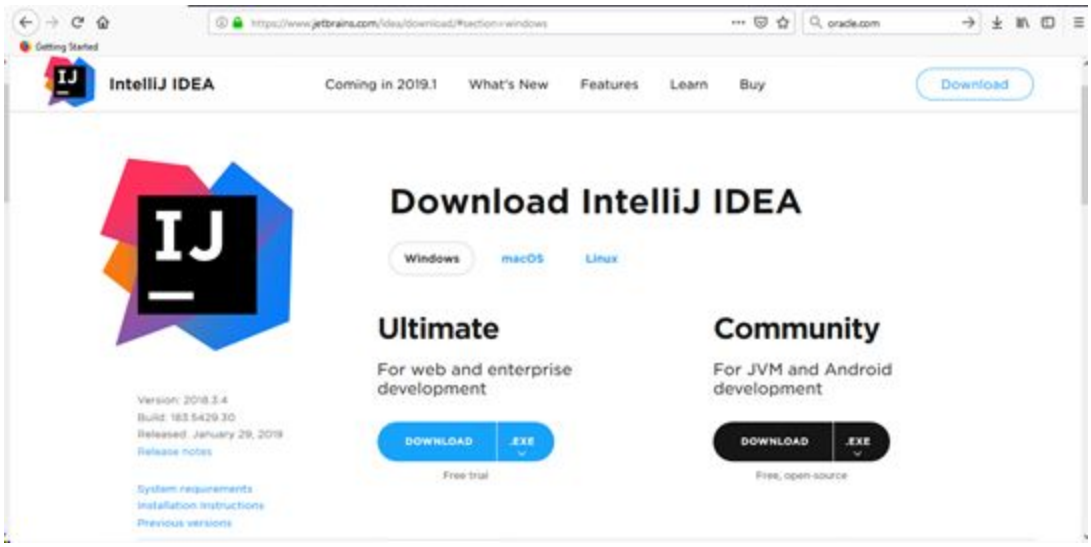


To Download IDE IntelliJ:

Follow this Link:

https://www.jetbrains.com/idea/download/#section=windows

Download Community Version

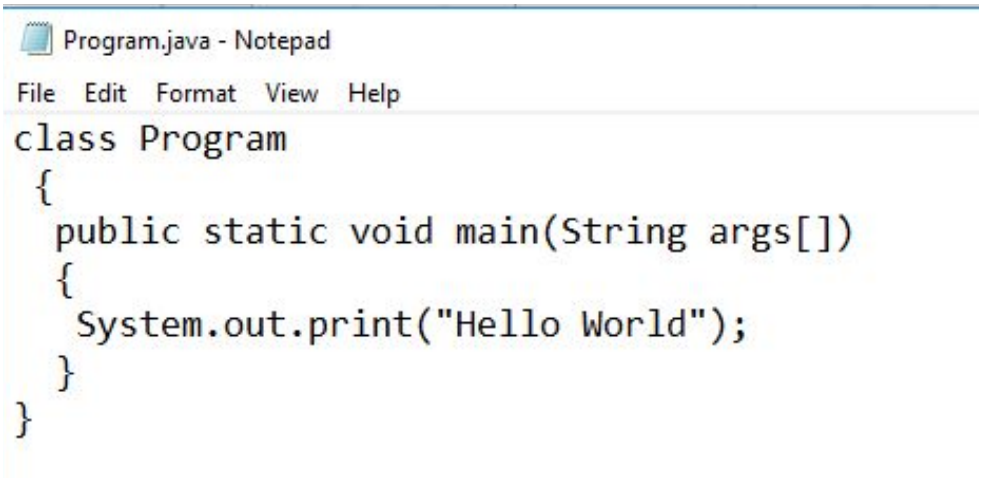


Walkthrough Task

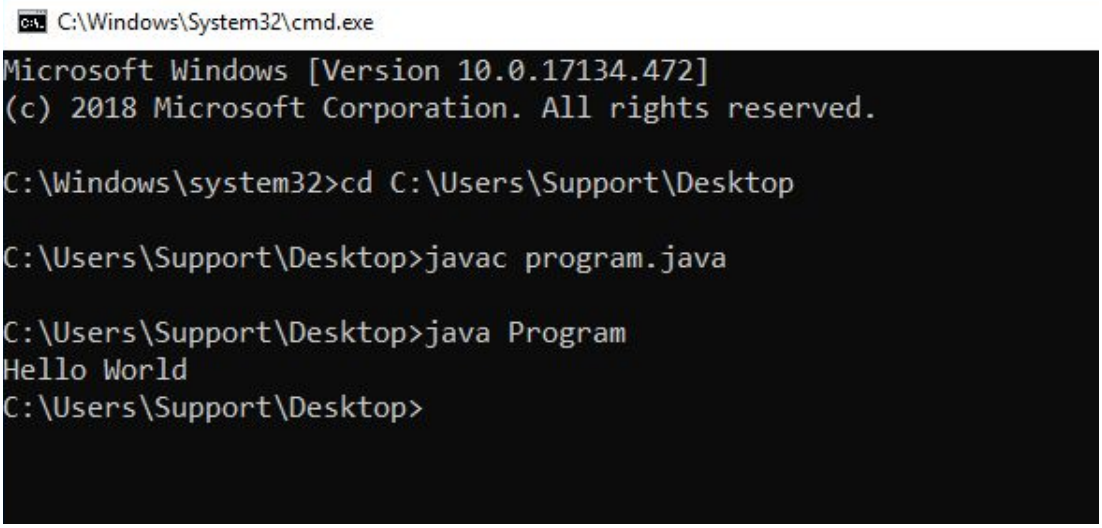
1. Writing a Simple Java Program

- Using Text Editor

1. **Create a source file -** A source file contains code, written in the Java programming language, that you and other programmers can understand. You can use any text editor to create and edit source files. Run notepad and enter your code. Save this file with the .java extension.



2. **Compile the source file into a .class file -** Open Command Prompt and enter **javac *filename.*java.** The Java programming language *compiler* (javac) takes your source file and translates its text into instructions known as *bytecodes*. Next, enter **java *ClassName.*** The Java application *launcher tool* (java) uses the Java virtual machine to run your application.



Print Functions:

println() : Prints the data normally then add a new line after it.

print(): Print will its parameters displayed in the command window, and the output cursor position after the last character display.

printf(): Printf is formatted output form.

Example:

```
public class demo {
public static void main(String args[])
  {
      int i=2;
      float j= 3f;
      System.out.print("The value of I is :"+ i);
      System.out.println( "The value of J is :"+ j);
      System.out.printf("The value of I is %d, J value is %f", i,j);

  }
}
```

Output

```
"C:\Program Files\Java\jdk-11.0.1\bin\java.exe"
The value of I is :2The value of J is :3.0
The value of I is 2, J value is 3.000000
Process finished with exit code 0
```

Enter a New line

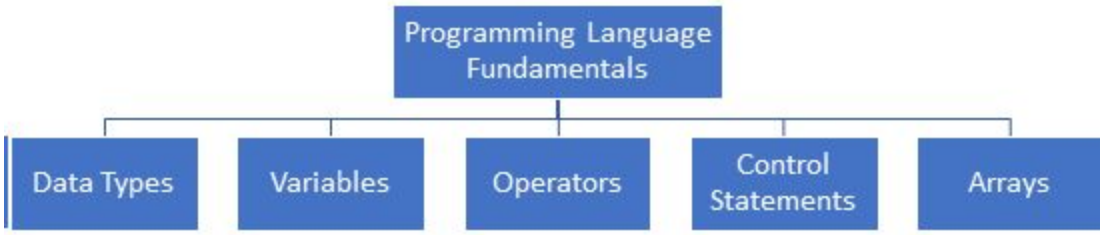
- **\r\n**, **%n, \n** are used to enter a new line.
- To enter a line independent of platform **System.lineSeparator()** is used.

Example:

```
System.out.print("The value of I is :"+ i+"\r\n");
System.out.println( "The value of J is :"+ j+"\n ***************");
System.out.printf("%d is an integer value,"+System.lineSeparator()+"%f is a float value %n ********",
i,j);
```

```
The value of I is :2
The value of J is :3.0
 ***************
2 is an integer value,
3.000000 is a float value
 ********
Process finished with exit code 0
```

Fundamentals:



Types, Values and Variables

The Java programming language is a statically typed language, which means that every variable and every expression has a type that is known at compile time.

The Java programming language is also a strongly typed language, because types limit the values that a variable can hold or that an expression can produce, limit the operations supported on those values, and determine the meaning of the operations. Strong static typing helps detect errors at compile time.

The types of the Java programming language are divided into two categories: primitive types and reference types. The primitive types are the Boolean type and the numeric types. The numeric types are the integral types byte, short, int, long, and char, and the floating-point types float and double. The reference types are class types, interface types, and array types.

The Kinds of Types and Values

There are two kinds of types in the Java programming language: primitive types and reference types. There are, correspondingly, two kinds of data values that can be stored in variables, passed as arguments, returned by methods, and operated on: primitive values and reference values.

Type:

Primitive Type

Reference Type

Primitive data types are the basic data types. These are predefined types of data, which are supported by Java. These data types are also called intrinsic or built-in types. The non-primitive data types in Java are objects and arrays. These non-primitive types are often called "reference types".
Below are the primitive data types in Java.

Data Type	Size
boolean	1 bit (true or false, by default false)
char	16 bits

Range of numeric data types in Java

Type	Size	Range
byte	8 bits	-128 .. 127
short	16 bits	-32,768 .. 32,767
int	32 bits	-2,147,483,648 .. 2,147,483,647
long	64 bits	-9,223,372,036,854,775,808 .. 9,223,372,036,854,775,807
float	32 bits	$3.40282347 \times 10^{38}$, $1.40239846 \times 10^{-45}$
double	64 bits	$1.7976931348623157 \times 10^{308}$, $4.9406564584124654 \times 10^{-324}$

Example:

```
public class demo {
public static void main(String args[])
 {
      int i=2;
      float j= 3f;
      byte  k=23;
      short l=43;
      long o=34543334;
      double d=4423.4323;
      char c='^';
      String s ="abc";
```

```
        boolean flag=true;
        System.out.println("i is int="+i);
        System.out.println("j is float="+j);
        System.out.println("k is byte="+k);
        System.out.println("l is short="+l);
        System.out.println("o is long="+l);
        System.out.println("d is double="+d);
        System.out.println("c is char="+c);
        System.out.println("flag is boolean="+flag);
 }
}
```

Operators

Operator in java is a symbol that is used to perform operations. For example: +, -, *, / etc. There are many types of operators in java which are given below:

Simple Assignment Operator

```
=     Simple assignment operator
```

Arithmetic Operators (binary operators)

```
+     Additive operator (also used
      for String concatenation)
-     Subtraction operator
*     Multiplication operator
/     Division operator
%     Remainder operator
```

Example:

```
public class demo {
public static void main(String args[])
 {
  int num1=3;
  int num2=2;
  float result=(float) num1/num2;
      System.out.println("Addition"+(num1+num2));
      System.out.println("Substraction"+(num1-num2));
      System.out.println("Multiplication"+(num1*num2));
      System.out.println("Division"+(num1/num2));
```

```
        System.out.println("Dision"+result);
        System.out.println("Modulus"+(num1%num2));
    }
}
```

Output:

Addition=5
Substraction=1
Multiplication=6
Division=1
Dision=1.5
Modulus=1

Unary Operators

```
+     Unary plus operator; indicates
      positive value (numbers are
         positive without this, however)
-     Unary minus operator; negates
      an expression
++    Increment operator; increments
         a value by 1
--    Decrement operator; decrements
      a value by 1
!     Logical complement operator;
      inverts the value of a boolean
```

Example:

```
System.out.println(--num1);
System.out.println(++num1);
System.out.println(num2--);
System.out.println(num2++);
System.out.println(num2);
```

Equality and Relational Operators

```
==    Equal to
!=    Not equal to
>     Greater than
>=    Greater than or equal to
<     Less than
<=    Less than or equal to
```

Conditional Operators

```
&&    Conditional-AND
||    Conditional-OR
?:    Ternary (shorthand for
        if-then-else statement)
```

Type Comparison Operator

```
instanceof      Compares an object to
                a specified type
```

instanceof is used to check if an object *is* an *instance* of a class, an *instance* of a subclass, or an *instance* of a class that implements a particular interface.

Bitwise and Bit Shift Operators

`~` `Unary bitwise complement :` inverts a bit pattern; it can be applied to any of the integral types, making every "0" a "1" and every "1" a "0". For example, a byte contains 8 bits; applying this operator to a value whose bit pattern is "00000000" would change its pattern to "11111111".

```
<<    Signed left shift
>>    Signed right shift
>>>   Unsigned right shift
&     Bitwise AND
^     Bitwise exclusive OR
|     Bitwise inclusive OR
```

Taking Input From the User

Using Buffered Reader Class

This is the Java classical method to take input, Introduced in JDK1.0. This method is used by wrapping the System.in (standard input stream) in an InputStreamReader which is wrapped in a BufferedReader, we can read input from the user in the command line.

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
public class Test
{
   public static void main(String[] args) throws IOException
```

```
    {
        //Enter data using BufferReader
        BufferedReader reader =
                new BufferedReader(new InputStreamReader(System.in));

        // Reading data using readLine
        String name = reader.readLine();

        // Printing the read line
        System.out.println(name);
    }
}
```

Note: To read other types, we use functions like Integer.parseInt(), Double.parseDouble(). To read multiple values, we use split().

Example:

```
int num1=Integer.parseInt(reader.readLine());
System.out.println(num1);
```

Using Scanner Class

This is probably the most preferred method to take input. The main purpose of the Scanner class is to parse primitive types and strings using regular expressions, however it is also can be used to read input from the user in the command line.

```
import java.util.Scanner;

class GetInputFromUser
{
    public static void main(String args[])
    {
        // Using Scanner for Getting Input from User
        Scanner in = new Scanner(System.in);

        String s = in.nextLine();
        System.out.println("You entered string "+s);

        int a = in.nextInt();
        System.out.println("You entered integer "+a);

        float b = in.nextFloat();
        System.out.println("You entered float "+b);
    }
```

}

Practice Tasks

Task 1

The following program will not compile because the lines have been mixed up.

```
System.out.print("Success\n");
}
public class Success
{
System.out.print("Success\n");
public static void main(String[] args)
System.out.print("Success ");
}
// It's a mad, mad program.
System.out.println("\nSuccess");
{
```

When the lines are arranged properly, the program should display the following output on the screen:

Program Output

```
Success
Success Success

Success
```

Rearrange the lines in the correct order. Test the program by entering it on the computer, compiling it, and running it.

Task 2

Write statements that do the following:

a) Declare a char variable named `letter`, and another named `input`.
b) Assign the letter 'A' to the `letter` variable.
c) Assign the input from user to the `input` variable
c) Display the contents of both variables.

Task 3

Suppose you earn Rs. 15,000 per month and you are allowed to contribute a portion of your gross monthly pay to a retirement plan. You want to determine the amount of your pay that will go into the plan if you contribute 5 percent, 8 percent, or 10 percent of your gross wages. Write a Java program to make this determination.

Task 4

Suppose a retail business sells an item that is regularly priced at Rs.500, and is planning to have a sale where the item's price will be reduced by 20 percent.
You have been asked to write a program to calculate the sale price of the item.
To determine the sale price you perform two calculations:

- First, you get the amount of the discount, which is 20 percent of the item's regular price.
- Second, you subtract the discount amount from the item's regular price. This gives you the sale price.

Evaluation Tasks

Task 1

Land calculation

One acre of land is equivalent to 43,560 square feet. Write a program that calculates the number of acres in a tract of land with 389,767 square feet. *Hint: Divide the size of the tract of land by the size of an acre to get the number of acres.*

Task 2

Circuit Board profit

An electronics company sells circuit boards at a 40 percent profit. If you know the retail price of a circuit board, you can calculate its profit with the following formula:

$$Profit = Retail\ price \times 0.4$$

Write a program that asks the user for the retail price of a circuit board, calculates the amount of profit earned for that product, and displays the results on the screen.

Task 3

Simple Interest Calculation

SMD bank wants you to develop an application that calculates Simple Interest for their Car Loans.
The application should take Principal Amount, Rate (per year in percentage) and Time (in years) from user and display amount of Interest.

Formula to calculate simple Interest

$$I=P*R*T / 100$$

Where I is amount of interest, P is principal amount, R is rate of interest and T is time.

Task 4

Marks Sheet

Educate Coaching center conducts their monthly assessments and make the result manually. To make the process efficient, you have to write a program that takes name of student, marks for the subjects (4) and calculate Total marks, Percentage and Average marks per course.

Total=Sub1+Sub2+Sub3+Sub4
Percentage=Total/400
Average Marks=Total/4

Home Tasks

Task 1

Cookie calories

A bag of cookies holds 40 cookies. The calorie information on the bag claims that there are 10 servings in the bag and that a serving equals 300 calories. Write a program that lets the user enter the number of cookies he or she actually ate and then reports the number of total calories consumed.

Task 2

Stock commission

Kashaf bought 600 shares of stock at a price of Rs. 21.77 per share. She must pay her stockbroker a 2 percent commission for the transaction. Write a program that calculates and displays the following:

- The amount paid for the stock alone (without the commission)
- The amount of the commission
- The total amount paid (for the stock plus the commission)

Task 3

Energy Drink Consumption

A soft drink company recently surveyed 12,467 of its customers and found that approximately 14 percent of those surveyed purchase one or more energy drinks per week. Of those customers who purchase energy drinks, approximately 64 percent of them prefer citrus flavored energy drinks.
Write a program that displays the following:

- The approximate number of customers in the survey who purchase one or more energy drinks per week
- The approximate number of customers in the survey who prefer citrus-flavored energy drinks

Task 4

Stock Transaction Program

Last month Javed purchased some stock in Acme Software, Inc. Here are the details of the purchase:

- The number of shares that Javed purchased was 1,000.

- When Javed purchased the stock, he paid Rs.32.87 per share.
- Javed paid his stockbroker a commission that amounted to 2% of the amount he paid for the stock.

Two weeks later Javed sold the stock. Here are the details of the sale:

- The number of shares that Javed sold was 1,000.
- He sold the stock for Rs.33.92 per share.

Write a program that displays the following information:

- The amount of money Javed paid for the stock.
- The amount of commission Javed paid his broker when he bought the stock.
- The amount that Javed sold the stock for.
- The amount of commission Javed paid his broker when he sold the stock.
- Display the amount of profit that Javed made after selling his stock and paying the two commissions to his broker. (If the amount of profit that your program displays is a negative number, then Javed lost money on the transaction.)