

**University of Bahrain**  
**College of Information Technology**  
**Department of Computer Science**  
**ITCS111/ITCS113**  
**Introduction to Computer Programming**

## 1 Basic Problems

1. Write a program that converts a distance measure from inches to cents. Note that one inch is equal to 2.54 centimeters?
2. The area of a *triangle* is computed by taking half the product of its base and height.

$$\text{Area} = \frac{1}{2} \times \text{base} \times \text{height}$$

Write a program that reads the base and height of a triangle and displays the area.

3. Students are awarded points toward their grades based upon the addition of the average of three quizzes ( $Q_1, Q_2, Q_3$ ), the midterm exam ( $MT$ ), and the final exam ( $FINAL$ ). Quizzes are marked out of 5, the midterm is out of 30, and the final is out of 70. Compute the total points using a Java program and print the result out of 100 according to the following percentages: Quizzes: 25%, Midterm: 35%, and Final: 40%.
4. Write a program that reads a student name followed by his/her id. The program should then display on screen the student's name followed by his/her academic year. For example, the academic year for the id 20102323 is 2010.
5. Write a program that calculates the energy needed to heat water from an initial temperature to a final temperature. Your program should prompt the user to enter the amount of water in kilograms and the initial and final temperature of the water. The formula to compute the energy is

$$Q = M \times (\text{final temperature} - \text{initial temperature}) \times 4184$$

Where  $M$  is the weight of water in kilograms, temperatures are in degrees Celsius, and energy  $Q$  is measured in joules. Format your output to three decimal places.

Sample Input/Output

Enter water amount:   **55.5**  
 Enter initial temperature:   **3.5**  
 Enter final temperature:   **10.8**  
 The energy needed is 1695147.600

6. Write a Java program that will predict tomorrow's temperature given today's temperature readings. The program will first ask the user to enter two values: the temperature *median* value and *current* temperature reading. The program will calculate and display the expected temperature according to the following equation:

$$\text{Expected temperature} = 2 \times \text{median temperature} - \text{current temperature}$$

Format your output to two decimal places.

Sample Input/Output

Enter temperature median value and temperature reading:  
**39.61    36.92**  
 Expected Temperature: 42.30

7. Write a program that reads an **employee name** and **two amount of sales** in Bahraini Dinar (BD) that represent the total sales of the employee in each day over the weekend (Friday and Saturday). Your program is then required to print the employee name followed by the percentages of sales on Friday and Saturday using the below equations. Format the output so that all numbers are printed with 2 decimal places.

$$\text{Friday Percentage} = \frac{\text{Friday Sales}}{\text{Friday Sales} + \text{Saturday Sales}} \times 100$$

$$\text{Saturday Percentage} = \frac{\text{Saturday Sales}}{\text{Friday Sales} + \text{Saturday Sales}} \times 100$$

Sample Input/Output

Enter your name: *Mustafa Ebrahim*  
 Enter Friday and Saturday Sales: *1500.500 2000.780*  
 Sales percentages of Mustafa Ebrahim are:  
 Friday is 42.86% and Saturday is 57.14%.

8. Write a program that reads three floating point numbers  $N_1$ ,  $N_2$ , and  $N_3$  from the user. It is required to find and print the value of the following formula:

$$F = N_1 + \frac{N_1 \times N_2}{4} - N_3$$

9. Write a program to read three resistance values  $R_1$ ,  $R_2$ , and  $R_3$  in Ohms and compute their combined resistance  $R_c$  when they are arranged in parallel. The value of  $R_c$  is computed using the following formula.

$$R_c = \frac{1}{\frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_3}}$$

Your program should print the values of  $R_1$ ,  $R_2$ ,  $R_3$  and  $R_c$ .

For example, if  $R_1 = 2.3$  Ohms,  $R_2 = 6.21$  Ohms, and  $R_3 = 4.58$  Ohms, then  $R_c = 1.228269131$  Ohms.

10. One large chemical company pays its salespeople on a commission basis. The salespeople receive BD. 75/600 per week plus 9% of their gross sales for that week. For example, a salesperson who sells BD. 1890/— worth of chemicals in a week receives BD. 75/600 plus 9% of 1890/—, or a total of  $75/600 + 170/100 = 245/700$ . Develop a program that will input each salesperson's gross sales for last week and will calculate and display that salesperson's earnings?
11. Write a program that reads two currency codes, exchange rate from first currency to second and the amount of money to be converted. Your program should display the converted amount for each currency using the following equations:

First Currency Amount = Currency Amount  $\times$  Exchange Rate

Second Currency Amount = Currency Amount / Exchange Rate

Format the output to 4 decimal places.

Sample Input/Output

Enter currency codes: *USD BD*  
 Enter Exchange rate: *0.3774*  
 Enter the amount: *125.0*  
 USD 125.0000 = BD 47.1750  
 BD 125.0000 = USD 331.2136

12. Write a program that reads from the keyboard the following input:

- Person Full Name written in a single line.
- Person CPR number (9-Digits) written in the next.

Your program should find and print on screen the birth year of this person. The birth year can be extracted from the first two digits of the CPR. Assume all persons were born before the year 2000 and after the year 1910.

Sample Input/Output

Enter your name: *Jassim Ali Ahmed*  
 Enter your CPR: *740707573*  
 Jassim Ali Ahmed was born in year 1974

13. ABC Phone Company, Inc., charges for phone calls by *distance* (miles) and length of *time* (minutes). The cost of a call (in Bahrain Dinar) is computed as 30% of the *call weight*, where the call weight is computed by adding the time with 5% of the distance.

Design a JAVA program that reads the distance and length of time for three phone calls. The program must calculate and display the cost for each of the three calls and the total cost of all three calls.

A sample Input/Output of your program is shown below.

```

Call 1: How many miles and minutes?  3 20
Call 2: How many miles and minutes?  2 15
Call 3: How many miles and minutes?  6 4

Call 1 costs BD. 6.045
Call 2 costs BD. 4.53
Call 3 costs BD. 1.29

Total cost is BD. 11.865

```

14. The Perimeter, Surface Area, and Volume of an in-ground pool are given by the following formulas:

$$\begin{aligned}
 \text{Perimeter} &= 2(\text{length} + \text{width}) \\
 \text{Volume} &= \text{length} \times \text{width} \times \text{depth} \\
 \text{Surface Area} &= 2(\text{length} + \text{width}) \times \text{depth} + \text{length} \times \text{width}
 \end{aligned}$$

Using these formulas as a basis, write a JAVA program that accepts the length, width, and depth measurements and then calculates the perimeter, volume, and surface area of the pool.

15. The volume of oil stored in an underground 200-foot-deep cylindrical tank is determined by measuring the distance from the top of the tank to the surface of the oil. Knowing this distance and the radius of the tank the volume of oil in the tank can be determined using the formula:

$$volume = \pi \times radius^2 \times (200 - distance)$$

Using this formula, write a JAVA program that accepts the radius and the distance measures, calculates the volume of the oil in the tank, and displays the two input values and calculates the volume.

16. The curriculum of any bachelor degree program consists of credit points which must be completed by all students enrolled in the program. Write a program that will ask the user to enter his/her **name**, program **credit**, and **expected** credits to be completed yearly. The program will then calculate degree duration in years. The **duration** in years is calculated by:

$$\text{Degree duration in years} = \text{program credits} \div \text{expected credits to be completed yearly}$$

The program will display the student name, degree duration in years, and expected graduation year formatted similar to the below sample output. Expected graduation year equals the year 2015 plus degree duration.

Sample Input/Output

<i>Enter your name, program credits and expected credits:</i>	
Ali	130 30
Name:	Ali
Degree duration:	4 years
Expected graduation year:	2019

17. Write a Java program that will perform some calculations regarding a cyclist coasting on a road. The program will ask the user to enter the cyclist's initial speed  $V_{\text{initial}}$ , the duration (in minutes), and the final speed  $V_{\text{final}}$ . The program will then calculate the rate of acceleration using the formula:

$$\text{acceleration} = (V_{\text{final}} - V_{\text{initial}}) \div \text{duration}$$

Next, calculate how long it will take for the cyclist to stop (given the initial speed and the calculated acceleration.) The necessary formula is:

$$\text{time} = V_{\text{initial}} \div \text{acceleration}$$

Display the acceleration and time on screen formatted to the below samples.

Sample Input/Output

Enter initial speed, final speed and duration 18.0    36.0    4.0  Acceleration = 4.5 4.0 minutes to stop
---

18. Write a program that reads from the keyboard two integers representing hours and minutes. The program should convert the time (hours and minutes) to seconds. Your program should write the result on screen as shown in the sample I/O below. the sample below:

Sample Input/Output

Hours?    10 Minutes?    15 Time in Seconds = 36900
---

19. The manager of a football stadium wants you to write a program that calculates the total ticket sales for a game. There are four types of tickets – Box, Sideline, Premium, and general Admission. Data is stored as shown below:

250	5750
100	28000
50	35750
25	18750

The first line indicates the ticket price is \$250 and that 5750 tickets were sold at that price. Output the number of tickets sold and the total sale amount. Format your output with two decimal places.

20. Write a program that reads three salaries from the keyboard and displays their average on screen formatted with 3 decimal places as shown below in the sample I/O.

**Sample Input/Output:**

Salary1	.....456.400
Salary2	.....1230.520
Salary3	.....650.550
Average	.....779.157

21. Two employees in a company are up for special pay increase. Write a program that reads the last name, first name, current salary and percent pay increase for both employees. Your program should display their salaries after the increase.
22. Write a program that calculates and prints the monthly paycheck for an employee. The net pay is calculated after the following deductions:

- Medicare/Medicaid Tax: 2.75%
- Pension Plan: 6%
- Health Insurance: BD. 75

Your program should prompt the user to enter the *gross amount* and print the *net pay*. A sample Input/Output is shown below:

```

Enter gross amount (BD): 650

----- Paycheck -----
Gross amount:           650.000
Medicare/Medicaid Tax:  17.875

Pension Plan:           39.000
Health Insurance:       75.000
-----
Net Pay:                 518.125

```

## 2 Flow Charts

1. In a game center, a child can play the games if the child's age is above 5 years and the height is above 120 cm. **DRAW A FLOWCHART OF A PROGRAM** that will ask the user to enter the child age and height, then the program should print a message saying "Yes, You can Play" if the child can play the game; otherwise it should print a message "Sorry, you cannot play".

2. Draw a *flowchart* for a program that:

Prompts the user to enter from the keyboard three numbers  $a$ ,  $b$ , and  $c$  representing the sides of a triangle. Based on the length of  $a$ ,  $b$ , and  $c$ , print the type of the triangle:

- **Equilateral:** if the three sides are equal.
- **Not Equilateral:** if the three sides are not equal.

Afterwards, display on screen the triangle circumference which is equal to the sum of  $a$ ,  $b$ , and  $c$ .

3. Nasser currently runs a car rental dealership and wishes to recommend a car to renters based on temperature of the location they want to visit. Draw a flowchart for a program to *read* a temperature of a location and *print* the recommended type as shown in the table below:

Temperature)	Recommended Car
Below 60	Truck
From 60 to 80 )	SUV
Above 80	Convertible

4. Draw a flowchart for a program to do the following:

- (a) Prompts the user to enter a decimal **value** and a character **code**.
- (b) If **code** is equal to 'R', the program calculates and prints the area of a circle and **value** is considered as the radius. The area of a circle is calculated as  $PI \times radius^2$ , where  $PI$  is equal to 3.14159.
- (c) If **code** is 'Q', the program calculates and prints the area of a square and **value** is considered as the side. The area of a square is calculated as  $side^2$ .
- (d) Any other code, display an error message and end the program.

### Example:

If the input is 2.3 and R, then the output is 'The area of a circle is 16.619' and if the input is 2.3 and Q, then the output is 'The area of a square is 5.29'.

5. The selling price of an apartment depends on the area of the apartment and the floor number where the apartment is located.

Draw a *flowchart* for a program that:

- (a) Prompts the user to enter from the keyboard:
  - The area by square meter of the apartment.
  - The floor number where the apartment is located.
- (b) Calculates the total price of the apartment based on the apartment area and the floor number as shown below:

Area (square meter)	Price for square meter
Upto 150	BD. 500 for each square meter
Between 151 and 200 (inclusive)	BD. 450 for each square meter
Over 200	Display apartment not available

- (c) Calculates additional **Price By Floor** using the floor number of the apartment multiplied by BD. 1000.
- (d) Display the total price if applicable.

**Apartment 1:** Area 120 in Floor 0, the total price =  $120 \times 500 + 1000 \times 0 = 60,000$ .

**Apartment 2:** Area 160 in Floor 5, the total price =  $160 \times 450 + 1000 \times 5 = 77,000$ .

6. Draw a *flowchart* for a program that:

- (a) Prompts the user to enter from the keyboard the employee's **Salary** in BD.

- (b) Calculates the **Tax** as shown in the table below:

Salary Range in BD	Tax
Between 0 and 500	$0.05 \times \text{Salary}$
More than 500 and less than 1000	$0.08 \times \text{Salary}$
1000 or Above	$0.09 \times \text{Salary}$

- (c) Displays the value of **Tax**.  
 (d) Displays the value of **Net Salary = Salary - Tax**.

**Example:** If Salary is 600, then Tax is  $0.08 \times 600 = 48$  and Net Salary is  $600 - 48 = 552$ .

7. Suppose you want to buy some pizzas from Cool Pizza restaurant.

**Draw a flowchart for a program that:**

- (a) Prompts the user to enter from the keyboard the **size**, the **quantity** and **service** charge of a pizza.  
 (b) Displays an error message and exits the program if the size of the pizza is less than 6 inches.  
 (c) Calculates the **Total Price** as shown in the table below:

Size of Pizza	Total Price BD
6 to 10 inches	$2.5 \times \text{quantity}$
greater than 10 inches	$3.5 \times \text{quantity}$

- (d) Displays the value of **Net Price = Total Price + Service charge**.

**Example:** If the size of the pizza is 7 inches, the quantity is 5 and service charge is BD. 0.850, then the Total Price is  $2.5 \times 5 = 12.5$  and the Net Price is  $12.5 + 0.95 = 13.35$ .

8. Draw a *flowchart* for a program that:

- (a) Prompts the user to enter from the keyboard **ONE** of the following characters: **T** or **C** followed by two numbers  $x$  and  $y$ .  
 (b) Calculates the value of  $f$  as follows:

$$f = \begin{cases} x - y, & \text{if T is entered;} \\ \frac{x - y}{x + y}, & \text{if C is entered;} \\ \text{Display "Error",} & \text{If other char is entered.} \end{cases}$$

- (c) Displays on the screen: The entered character, the values of  $x$  and  $y$ , and the value of  $f$ .

9. The “Burger & Fries” restaurant is running a promotion in which delivery orders receive a discount while dine-in orders receive no discount. **Draw a flowchart** that displays the discount percentage. The program should first ask the user for two values: order **type** (delivery or dine-in), and the **number** of burgers to be ordered. The program will then determine the discount as follows:

- (a) **Dine-in** orders receive no discount.  
 (b) **Delivery** orders of up to two burgers receive 25% discount.  
 (c) **Delivery** orders of larger than two burgers receive 40% discount.

The program will select from above which discount percentage to display.

### 3 Selection Problems

1. Write a program to find the ideal weight for a given height. The height must be given in centimeters between 140 cm and 240 cm. The ideal weight is the height minus 100. Before calculating the height check that the input is valid?
2. ACE balloon shop sells balloons to customers at 10 fils each. If the customer buys 500 – 1000 balloons, the price is 8 fils each. If the customer buys over 1000 balloons, the price is 6 fils each. Write a program that reads the number of balloons to purchase and prints the amount of the bill?
3. A part time employee is paid BD. 2/000 per hour for the first 20 hours in a week. Additional hours are paid BD. 2/500 per hour. In addition the company deducts 5% of the total weekly income for health insurance. Find how much (s)he will be paid for any given number of hours?
4. Write a Java program that asks the user to input the weight of a canary bird in grams. The program should determine and display a message the category of the bird's weight according to the following table:

Money Spent (in BD)	Coupon Percentage
Less than 10	No coupon
From BD 10 to 60	8%
More than 60 to 150	10%
More than 150 to 210	12%
More than 210	14%

For example, if you spend BD 50, you will get a coupon worth eight percent of that amount. Write a program that calculates and prints the value of the coupon a person can receive based on groceries purchased.

#### Sample Input/Output

Please enter the cost of your groceries: 14  
You won a discount coupon of BD 1.12. (8% of your purchase)

5. A supermarket awards coupons depending on how much a customer spends on groceries. The following table shows the percent used to calculate the coupon awarded for different amounts spent.

Weight	Category
Between 50 Grams and 100 Grams (inclusive)	Healthy
Under 50 but above 20 OR above 100 but under 130	Critical
All other ranges	Endangered

6. A valid mobile number must consists of 8 digits. Write a program that reads a mobile number of type long and verify that it's exactly 8-digits. If the mobile number is not exactly 8 digits, then display an invalid message and quit the program. Otherwise, your program will print the provider name according to the following table.

Provider Name	Mobile number
Zain	starts with 36
Batelco	starts with 34 or 39
Viva	starts with 33
Invalid	otherwise

**Hint:** A number is exactly 8 digits if it is in the range 10000000 – 99999999.

7. Write a Java program that accepts a number followed by one space and then a letter. If the letter following the number is 'f' or 'F', the program is to treat the number entered as a temperature in degrees Fahrenheit, convert the number to the equivalent degrees Celsius, and print the result. If the letter following the number is 'c' or 'C', the program is to treat the number entered as a temperature in degrees Celsius, convert the number to the equivalent degrees Fahrenheit, and print the result. If the letter is neither 'f', 'F', 'c', nor 'C', the program is to print a message that the input is invalid and terminate. Note that  $C = \frac{5}{9}(F - 32)$ .

8. Write a Java program that accepts a number followed by one space and then a single word. If the word following the number is “mile” (ignore case), the program is to treat the number entered as a distance in miles, convert the distance to kilometers. If the word following the number is “kilometer” (ignore case), the program is to treat the number entered as a distance in kilometers, convert the distance to miles. Any other word the program prints an invalid message. Note that one mile = 1.609 kilometers.
9. Scores attained by a student in two exams are to be entered to a computer (double numbers between 0.0 and 100.0, inclusive). An indication of **Pass** or **Fail** is printed out after the two marks are entered. The criteria for passing are as follows:
  - A student passes if **both** examinations are passed where the pass mark for an individual subject is 60.
  - Additionally, a student may pass if only one subject is failed and the overall average is greater than or equal to 60.

Write a Java program that prompts the user to input two exam marks and prints whether the student has passed the course or failed.

10. Write a program to calculate and display the net price of a *pasta* based on two user inputs: **type** (C for chicken, M for Meat, or V for Vegetable) and the number of **appetizers**. the net price of the pasta is the sum of the following:
  - (a) Base price of the pasta is BD. 2.5 for chicken, BD. 2.5 for Meat, and BD. 2.1 for vegetables.
  - (b) Each appetizer is an additional BD. 1.2

Sample Input/Output

Enter pasta type: C for chicken, M for Meat, V for Vegetables:  
C  
Enter number of appetizers:  
2  
Net price is BD. 4.9

11. Write a program that reads two points on a line and another two points on another line. Your program should determine whether these two lines are *parallel* or they *intersect*. If they intersect, you need further to state whether they are *perpendicular* or not.  
  
Two lines are parallel if their slopes are equal, otherwise they intersect. Two intersected lines are perpendicular if the product of their slopes is -1. Use the formula below to compute the slopes of the two lines.  
  
Slope of  $(x_1, y_1)$  and  $(x_2, y_2)$  is  $\frac{y_2 - y_1}{x_2 - x_1}$ .
12. Write a Java program that asks the user to enter two float numbers and a character which represents a code for an arithmetic operation. The program performs arithmetic operations (addition, subtraction, multiplication, and division) on those two numbers depending on the value of code as follows:

Code	Arithmetic Operation
A or a	+
S or s	-
M or m	*
D or d	/

You must use the *switch* statement to determine the arithmetic operation. If the code character that is entered by the user is not one of those specified above, print a message saying “invalid code”.

13. A wholesale department is offering reduced prices on large orders of kitchen utensils. Write a Java program to calculate and display the net price of an order based on three user inputs: **Price** (float), **Quantity** (int) and **Payment** method (char: **A** for cash in advance and **D** for cash on delivery.) The net price of the order is calculated as follows:

Net price = (Price - Discount1 - Discount2) × Quantity

Discount1 = 4% of Price for Quantities of more than or equal to 1000, otherwise 2% of Price.

Discount2 = 5% of Price for cash in advance payment and no discount for cash on delivery.



Sample Input/Output

Enter Price, Quantity and Payment Method (A or D)  
 10    2500    A  
 The net price is 22750

14. The “*Burger and Fries*” restaurant is running a promotion in which delivery orders receive a discount while dine-in orders receive no discount. Write a program that displays the discount percentage. The program should first ask the user for two values: order **type** (delivery or dine-in), and the **number** of burgers to be ordered. The program will then determine the discount as follows:

- (a) **Dine-in** orders receive no discount.
- (b) **Delivery** orders of up to two burgers receive 25% discount.
- (c) **Delivery** orders of larger than two burgers receive a 40% discount.

The program will select from the above which discount percentage to display.

15. A triangle is a geometric shape in which no side can be larger than the sum of the other two sides. Write a Java program that:
- Prompts the user to enter from the keyboard 3 values of type float and each value must be larger than zero: x1, x2, and x3.
  - Prints a message indicating ”invalid entered values” if any of the entered values is NOT larger than zero.
  - Checks whether the entered values x1, x2, and x3 form a valid rectangle or not, and prints a proper message.
16. Write a program that asks the user for a month as a number between 1 and 12. The program should display a message indicating whether the month is in the first quarter, the second quarter, the third quarter, or the fourth quarter of the year. Following are the guidelines:

Month	Quarter
1, 2, 3	First
4, 5, 6	Second
7, 8, 9	Third
10, 11, 12	Fourth
other	Invalid Quarter

17. Write a JAVA program that determines the alert level for the COVID-19 based on the percentage of new cases out of PCR tests carried out as indicated in the following table:

Alert Level	Percentage of New Cases
Green	< 2%
Yellow	Between 2% and 5% (inclusive)
Orange	Between 5% and 8% (exclusive)
Red	8% or above

$$\% \text{ of new cases} = \frac{\text{number of new cases}}{\text{number of PCR tests}} \times 100$$

The program should ask the user to input the number of new cases and the number of PCR tests that were carried out. The program should display the percentage of new cases and the alert level.

Sample Input/Output

Enter the number of new cases: 280  
 Enter the number of PCR tests: 12000  
 Percentage of new cases is 2.333333333333335%  
 Alert Level is Yellow

## 4 Loops problems

1. Blood sugar is considered *normal* if its rate is less than 140 and greater than 70, it is considered *high* if its rate is greater than or equal to 140, and it is considered *low* if its rate is less than or equal to 70.

Write a Java program that asks the user to input the rate of blood sugar of each patient in a hospital (-1 to indicate the end of the data.) The program should calculate and print the number of patients with normal blood sugar, number of patients with high blood sugar, number of patients with low blood sugar, and the total number of patients entered by the user.

2. Write a Java program that asks the user to input a line of characters from the keyboard. Your program should count the number of alphabets (a-z, A-Z) and the number of non-alphabets (i.e digits, special characters, spaces, tabs, etc.) in the given input.

### Sample Input/Output

```
Enter a line of characters:
CSC 103

Alphabets are 3
Non-alphabets are 4
```

3. Write a program that asks the user to enter the number of times that they have run around a racetrack, and then uses a loop to prompt them to enter the lap time (in minutes) for each of their laps. When the loop finishes, the program should display the time of their fastest lap and their average lap time.

### Sample Input/Output

```
How many laps did you run? 4

Enter the time for lap 1: 4. 75
Enter the time for lap 2: 5.25
Enter the time for lap 3: 4.5
Enter the time for lap 4: 6.0

Fastest lap is 4.5 and the average
is 5.125
```

4. Write a Java program using a **LOOP** to print the date of all Sundays in a month, if the date of the first Sunday is given. First prompt the user to input the number of days in the month, then prompt the user to input the date of the first Sunday in that month. After that find all Sundays in the given month and output. **Assume all inputs are valid.**

Your program must follow the given Sample Input/Output.

### Sample Input/Output

```
Enter number of days in the month: 31
Enter the date of first Sunday: 3
All Sundays in the month will be: 3 10 17 24 31
```

5. In one of the introductory courses in a university, passing students are assigned a final grade of the uppercase letter **P** or lowercase **p** while failing students are assigned the letter **F** or **f**. Write a Java program that asks the user to enter grades (P/p or F/f) for 30 students, then calculates and displays both percentages of the passing and failing students.
6. A factory manufactures electricity cables, all of which are tested to find their Cable Factor. Cables are classified as either Accepted or Rejected as follows:

Cable Factor	Cable Quality
Greater than or equal to 4.5	Accepted
Otherwise	Rejected

Write a program that asks the user to enter the Cable Factor of several cables until the user enters a negative number to quit. The program should find and display:

- (a) The average cable factor for **Accepted** cables.
  - (b) The number of cable factors entered by the user.
7. Develop an algorithm for a simple game of guessing at a secret five-digit code. When the user enters a guess at the code, the program returns two values: the number of digits in the guess that are in the correct position and the sum of those digits. For example, if the secret code is 53840, and the user guesses 83241, the digits 3 and 4 are in the correct position. Thus, the program should respond with 2 and 7. Allow the user to guess a fixed number of times. **Remark.** To generate a random number in the range from  $A$  to  $B$  use the below line of code:

```
int num = (int) (Math.random() * (B - A + 1)) + A;
```

For guessing a number that is exactly five digits use  $A = 10000$  and  $B = 99999$  in the above line of code.

8. One way of hiding a digit (0 to 9) is to include it in the right-most digit of any number. For example, the number 9423 hides the digit 3 and 890 hides the digit 0. Write a program that reads a set of integers for a hidden number and stops reading when the user enter a negative number. Your program is required to print the sum of the digits of the hidden number.
9. Write a program that reads a positive integer number of any digit size and do the following:
  - (a) Print the sum of its digits.
  - (b) Print the average of its digits.
  - (c) Print the leftmost digit.
  - (d) Print the maximum digit.

If the input number is negative, convert it to positive. Here is a sample Input/Output of your program:

```
Enter a positive integer: 1246
The Sum is 13
The average is 3.25
The leftmost digit is 1
The maximum is 6
```

10. Wind speed is classified according to their velocity into *not a strong wind*, *strong wind*, *gale wind*, *whole gale wind* and *hurricane*. Write a program that asks the user to enter a wind speed which is an integer number and then display the wind category depending on that value. Your program should continue asking for a wind speed and displaying its category until the user enters a negative value. The wind speed is classified as follows:

Wind Speed	Category
Below 25	Not a strong wind
25-38	Strong wind
39-54	Gale
55-72	Whole gale
Above 72	Hurricane

### Sample Input/Output

```
Enter wind speed> 45
Gale
Enter wind speed> 19
Not a strong wind
Enter wind speed> -1
```

11. Write a Java program that asks the user to input the age and gender (M for male, F for female) of 10 rabbits. Your program should count the number of female and male rabbits and print the results on the screen.

#### Sample Input/Output

```
Enter the age and gender of 10 rabbits:
3  M
1  M
  ⋮
5  M

Male Rabbits = 4
Female Rabbits = 6
```

12. The admission office of the university is requested to write a program that reads records for multiple students. Each record consists of first name (string) followed by his/her GPA (double) followed by a letter code, either F or M (char). The letter code F is used for female students and M for male students. Assume the number of records is unknown and the program stops reading when the word 'stop' is entered as the student name. Write a program that reads the first name, the GPA and the letter code for every student and displays on screen the following:
- The number of students named AHMED.
  - The number of students whom GPAs are above 3.0.
  - The name that has the maximum GPA.
  - The average GPA for female and male students separately.
13. After graduation from high school, students with percentages above or equal to 90% are getting scholarships and students with percentages below 70% but not below 60% must register for the ENGLISH class. Write a Java program to do the following:
- Prompt the user to enter the number of students. If the user entered a negative or zero number of students, then print an invalid message and ask the user to try again (see the sample input/output).
  - Ask the user to enter percentages (double) for all students.
  - Display how many students have scholarships and how many students registered for the ENGLISH class.

#### Sample Input/Output

```
How many students? -3
Invalid input. Try again? 6
Enter percentages for 6 students:
85.0
68.0
92.5
61.0
95.5
64.0

2 have scholarship and 3 registered ENGLISH
```

14. Write a Java program to generate a user's daily average number of walking steps for a given month of the year. To do so, write the appropriate code for each of the following steps:
- First the program should prompt the user to enter the month's number (from 1 to 12) as an integer type. Assume all input is valid.

- (b) Use switch case to determine the number of days in the given month (Suppose that month 2 has 28 days, months 1,3,5,7,8, 10,12 have 31 days and the other months have 30 days).
- (c) Use a counter-based loop (for-loop or while-loop) to ask the user to enter the number of steps walked in each of the days of the entered month.
- (d) Calculate and display the daily average number of walked steps as follows:

$$\text{Daily Average of walked steps} = \frac{\text{Total steps walked in a given month}}{\text{Number of days of a given month}}$$

#### Sample Input/Output

Input the month: 2  
 Input the number of steps for each of the 28 days of month 2:  
 227 1232 119 12002 1222 ... 222  
 Daily Average of Walked Steps = 4732.28322

15. A triathlon is a multiple stage sports competition, where athletes compete for fastest overall course completion time. Write a Java program that asks the user to enter the name and time for 50 athletes, then prints the name and time of the winner. All times are measured in minutes (int). The winner has the lowest time. Assume all inputs are valid.

#### Sample Input/Output

Enter the name and time for 50 athletes:  
 Mike 274  
 John 207  
 Bob 301  
 ...  
 Alistair 206  
 Comac 244  
 The winner is Alistair with completion time of 206 minutes

16. Write a JAVA program that asks the user to input the jackets quality in any letter case (HQ for High Quality, LQ for Low Quality) and their prices in a factory, until the user enters a QUIT in any letter case to stop. Your program should count the number of HQ and LQ jackets, as well as, the highest HQ price of a jacket, then print the results on the screen. **Your program should ignore any unknown jacket quality.**

#### Sample Input/Output

Enter the quality of the jackets followed by their prices (quit to stop):  
 LQ 100.5  
 HQ 382.99  
 Lq 411.05  
 hq 277.78  
 sd 120.5  
 HQ 299.5  
 Quit  
 Number of HQ jackets = 3  
 Number of LQ jackets = 2  
 Highest HQ Price BD = 382.99

17. Assume that we have records of bank clients each consists of client **name** and his/her saving **money** in the bank. Write a program that reads all client records and stops reading when the word 'stop' is entered as the client name. The program should display on screen the total saving money in the bank and the name of the client who has the highest saving money.

**saving.txt**

```

Enter client records:
Ahmed 500
Mohammed 600
Ali 3000

Total Saving amount = 5100
Ali has the highest saving money 3000
  
```

18. Consider records of *transactions* each consist of a **username** (string), **amount** (float), and **membership number** (int). Write a Java program that reads records of multiple transactions and displays on screen the number of users, the total amount and the highest amount. Reading records of transactions should stop when the username "user999" is entered by the user.

**Sample Screen Output**

```

Enter transaction records:
user001 12.5 12345
user002 5.2 87542
user003 33.1 93568
...
user143 23.56 67332
user999

Number of users: 54
Transactions total amount: 2658.7
The highest amount: 280
  
```

19. Consider records of *employees* each consists of employee's first name (string), number of hours worked (int) and hourly rate in Bahraini Dinar BD. (double). Write a program that reads from the user multiple records for multiple employees and displays the following on screen:
- (a) the salary for each employee (Salary = Hours Worked × Hourly rate.)
  - (b) the number of employees
  - (c) the highest salary with employee name.

Reading records of employees should stop when the user enter 'stop' for the employee name. Format the salary to 3 decimal places. Sample I/O of the program is shown in the example below.

**Sample Output**

```

Enter name, hours, and rate:
Emad 45 3.5
Salary is BD. 157.500
Enter name, hours, and rate:
Amina 77 9.0
Salary is BD. 693.000
...
Enter name, hours, and rate:
Faisal 60 4.75
Salary is BD. 285.000
Enter name, hours, and rate:
stop

There are 63 employees .
Amina has the highest salary of BD. 693.000
  
```

20. Special promotion was given for mobile subscribers that offers 300 free minutes call and 30 free SMS messages for only BD. 6. Any additional call minute is charged for BD. 0.015 per minute and additional SMS message is charged for BD. 0.010 per SMS message. Each mobile phone subscriber has a record that consists of mobile phone number, number of call minutes used, and number of SMS messages sent. Write a program that reads multiple records for multiple subscribers. Stop reading when the user enter negative or zero for the phone number. The program should print on screen the total bill for each subscriber and the telephone number of the lowest bill.

$$\text{Total Bill} = \text{BD. 6} + \text{Extra Minutes Charge} + \text{Extra SMS charge}$$

#### Sample Output

```

Enter phone number, calls, and messages:
39661122  280  21
Bill is BD. 6.000
Enter phone number, calls, and messages:
36664444  430  37
Bill is BD. 8.020
...
Enter phone number, calls, and messages:
33331234  320  10
Bill is BD. 6.300
-1

Telephone number 39661122 has the lowest bill

```

21. The grade level of undergraduate college students is typically determined according to the following schedule:

Number of Credits Completed	Grade Level
Less than 32	Freshman
32 to 63	Sophomore
64 to 95	Junior
96 or more	Senior

Number of Credits is valid if it is from 0 to 133, and any other value is considered invalid.

Using this information, write a Java program that accepts the number of credits of each student in a college (negative number to stop) and determines the **percentage** of Freshmen students, Sophomore students, Junior students, and Senior students.

22. A student is expected to graduate if he/she has a GPA greater than or equal to 2.0 and has completed at least 95 credit hours. Write a program the reads student ID, GPA, and credit hours completed for multiple students. The program should stop reading when a negative ID is entered and print the number of students expected to graduate.

#### Sample Input/Output

```

Enter student's id, GPA, and hours (negative to stop)
20081234  3.3  87
20081235  2.3  99
20092351  1.9  50
20071236  3.7  101
20081237  1.99 96
20108765  3.2  45
20073254  2.8  112
-1

Expected to graduate are 4 students.

```

23. Write a Java program that reads two integers *num1* and *num2*. If *num1* is greater than *num2*, then swap (exchange) them. Next, your program must display all numbers from *num1* to *num2* on a single line such that every line contains the number followed by all its divisors (see sample input/output below.)

Sample Input/Output

Enter two numbers: 20 25						
Number	Divisors					
20	2	4	5	10	20	
21	3	7	21			
22	2	11	22			
23	23					
24	2	3	4	6	8	12 24
25	5	25				

24. Write a Java program that reads two positive integers *num1* and *num2* (assume *num1* ≤ *num2*). Then, the program will display all perfect numbers between *num1* and *num2*. A positive integer is called a perfect number if it is equal to the sum of all of its positive divisors, excluding itself. For example, 6 is the first perfect number because  $6 = 3 + 2 + 1$ . The next is  $28 = 14 + 7 + 4 + 2 + 1$ . **Hint:** use modulus (%) to test for a divisor of a number.

Sample Input/Output

```
Input the 2 numbers: 6 1001
Perfect Numbers are:
6
28
496
```

25. Write a Java program using **Nested Loops** to print the factorials of all numbers between a range of two integers namely first and second. The first/second integers can be zero or any positive numbers and the first should be always less than or equal to second. Appropriate error message should be displayed in case of invalid inputs.

Note that a factorial of any integer can be calculated as follows:

Factorial of 0 is 1

Factorial of 1 is 1

Factorial of 2 is  $2 \times 1 = 2$

...

Factorial of 5 is  $5 \times 4 \times 3 \times 2 \times 1 = 120$

...

Factorial of *N* is  $N \times (N - 1) \times (N - 2) \times \dots \times 1$

Your program must follow the given Sample Input/Output.

Sample Input/Output

```
Enter the first number: 4
Enter the second number: 7
Factorial of 4 is 24
Factorial of 5 is 120
Factorial of 6 is 720
Factorial of 7 is 5040
```

26. Write a program that reads from the keyboard a positive integer *M* and generates a triangle of numbers (as shown below). You must make sure the number is entered is positive before making the triangle. For example, if the input for *M* is 6, the output triangle should be

```
6 5 4 3 2 1
5 4 3 2 1
4 3 2 1
3 2 1
```



```

2 1
1

```

27. Write a program that reads two integers *num1* and *num2*. If any of these integers is negative convert it to positive and if  $num1 > num2$ , swap *num1* and *num2*. The output of your program is as shown below:

```

Enter first number: 3
Enter second number: 7

```

```

3      6      9      ...    30
4      8      12     ...    40
5      10     15     ...    50
6      12     18     ...    60
7      14     21     ...    70

```

28. Write a Java program that prompts the user to input an integer greater than 0 which represents the size/height of the triangle shape. The program should print the following triangle shape according to the user input by alternating '!' and '+' characters. Invalid error message must be displayed if the size is less than 1.

**Example:** if the size of the shape = 4, then the program should output:

```

      !
    ! + !
  ! + ! + !
! + ! + ! + !

```

29. Write a program that reads an integer number *N*. If the number is negative convert it to positive. The program is required to produce a shape consisting of \* and # characters of *N* lines. Below is shown examples on sample output for some values of *N*.

```

N = 1      N = 2      N = 3      N = 4
*#          **#       ***#      ****#
           **#       ***#      ****#
           **#       ***#      ****#
                           ****#
                           ****#

```

30. Write a Java program that asks the user to enter a **positive** integer  $N \leq 15$ . The program should check that *N* is valid and repeatedly asks the user to enter *N* if *N* is not valid. The program then should print the following shape according to the value of *N* by alternating \* and + characters.

#### Examples

Example 1	Example 2
<p><i>Enter a positive integer N between 1-15: 5</i></p> <pre> *+*+*+*+* *+*+*+* *+*+* *+* * </pre>	<p><i>Enter a positive integer N between 1-15: 5</i>  <i>Enter a positive integer N between 1-15: -3</i>  <i>Enter a positive integer N between 1-15: 3</i></p> <pre> *+*+* *+* * </pre>

31. Write a program that uses **nested for-loops** to display a square of # and + characters as shown in the examples below. The program should ask the user to enter a positive odd number *N* greater than or equal to 3 representing the square size in terms of number of rows and columns. Appropriate error message should be displayed if the user input for *N* is invalid.

## Examples

If $N = 3$ then the shape should be as follows:	If $N = 5$ then the shape should be as follows:	If $N = 7$ then the shape should be as follows:
<pre> #+# +++ #+# </pre>	<pre> ##+## ##+## ##+## +++++ ##+## ##+## </pre>	<pre> ###+### ###+### ###+### +++++++ ###+### ###+### ###+### </pre>

32. Write a Java program that uses **nested for-loops** to display a shape of stars (see examples.) The program should ask the user to enter an integer **n** representing the number of rows/columns in a square. The number **n** should be odd and larger than 2. If the user input for **n** is invalid, then display an appropriate message and exit the program. The program should display the shape of number 7 reversed (flipped to the left.)

## Examples

If $N = 3$ then the shape is:	If $N = 5$ then the shape is:	If $N = 7$ then the shape is:	If $N = 4$ then the shape is:
<pre> ***  *   * </pre>	<pre> *****  *   *    *     * </pre>	<pre> *****  *   *    *     *      *       * </pre>	Incorrect Number.

33. Write a Java program that displays a shape representing the capital letter of Z as shown in the example below. The program should ask the user to enter an odd integer number greater than 2 representing the size of the letter in terms of number of rows and columns. Appropriate error message should be displayed and quit the program if the user input is invalid.

## Sample Input/Output

Enter odd number > 2: <b>6</b> Invalid number. Bye bye.
Enter odd number > 2: <b>5</b>  <pre> ##### # # # ##### </pre>

34. Repeat Problem (33) for the below letters.

## Sample Input/Output

Enter odd number > 2: <b>5</b>  <pre> #  # #  # ##### #  # #  # </pre>
--

## Sample Input/Output

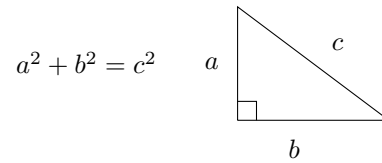
Enter odd number > 2: <b>5</b>  <pre> ##### # # # ##### </pre>
--

## Sample Input/Output

Enter odd number > 2: <b>5</b>  <pre> ##### # # # ##### </pre>
--

## 5 Methods

- (a) Write a method named **isRightTriangle** that takes 3 sides (double) of a triangle as parameters. The method should return **true** if the sides represent a right triangle, otherwise, it returns **false**.



- (b) Write a program that prompts the user to enter three sides of a triangle (double) and displays if the sides represent a right triangle or not by using the method defined in part (a). A sample of the The input/output screen is shown below.

Sample input/Output 1	Sample input/Output 2
Enter 3 sides: <b>3.0 4.0 5.0</b>	Enter 3 sides: <b>2.0 5.0 6.0</b>
<b>Yes, it is a right triangle.</b>	<b>No, it is not a right triangle.</b>

- The distance between points with coordinates  $(X_A, Y_A)$  and  $(X_B, Y_B)$  is given by

$$DISTANCE = \sqrt{(X_A - X_B)^2 + (Y_A - Y_B)^2}$$

You are given the coordinates of three points: point1  $(X_1, Y_1)$ , point2  $(X_2, Y_2)$ , and point3  $(X_3, Y_3)$ . Write a program to read the coordinates. Next write a method that calculates the distance between any two points. Use this method to find and print the distance DIST12 between points 1 and 2, distance DIST1 between points 1 and 3, and distance DIST23 between points 2 and 3.

**Note:** Use the methods `Math.sqrt(x)` and `Math.pow(x,y)` in the `Math` class for  $\sqrt{x}$  and  $x^y$ , respectively. See page 446 in your textbook.

- Write a method `trapezoid(int start, int end)` that generates a trapezoid of numbers as shown below. *start* and *end* are integers from 0 to 9.
  - If *start* is not between 0 to 9 (inclusive) set it to 0.
  - If *end* is not between 0 to 9 (inclusive) set it to 9.
  - If *start* > *end*, swap *start* and *end*.

For example, if the value of *start* is 4 and *end* is 7, the trapezoid method should produce an output as:

```
4444
55555
666666
7777777
```

- (A) Write a method that accepts a train code of type character (lower or upper case) and returns the corresponding train type according to the following table:

Train Code	Train Type
S or s	Steam
D or d	Diesel
E or e	Electric
Others	Invalid

- (B) Write a program that prompts the user to enter 10 train codes (char) and displays their corresponding train types by using the method defined in (A). The input/output of the program should be displayed as follows:

```
Enter 10 train codes:
S
This code represents Steam Train Type
M
This code represents Invalid Train Type
d
This code represents Diesel Train Type
.....
```

5. A company needs to compute the bonus earned by its salespeople. This bonus is computed from the total sales of the salesperson, using the following table:

Total Sales	Bonus Percent
$\text{sales} < \$1000$	1%
$\$1000 \leq \text{sales} < \$2000$	2%
$\$2000 \leq \text{sales} < \$2500$	3%
$\$2500 \leq \text{sales}$	5%

Write a method to compute the sales bonus and use it in a complete program.

6. Write a Java program to compute the surface area of a cylinder, given the radius (R) and height (H). The program must ask the user to enter R and H. It then prints the surface-area with a proper message in the main program.

Define a method to compute the surface-area and let `main()` call it. Clearly show the method *prototype*. This method takes the radius R and height H as arguments and returns the surface area. The formula for computing the surface-area of a cylinder is  $2\pi R(H + R)$ , where  $\pi = 3.141592654$ .

7. Write a method called `isValidCode()` that takes an integer number and returns 1 if the number represents valid code and 0 otherwise. Valid code is determined according to the following rules:

- The number length is at least 3 digits and at most 5 digits.
- All the digits must be between 1 and 5 (inclusive).
- The sum of the digits must be odd number.

For example, the number 324 is valid because it's 3 digits long, all digits are between 1 and 5, and the sum of the digits is 9, which is odd. However, the number 3256 is not valid because it consists of the digit 6 which is not in the range 1 to 5. Also the the number 114 is not valid because the sum is 6 which is not odd.

8. Write a method whose input is an integer. The method is to return an integer whose digits are reversed from the input integer. Thus, if 1432 is the input to the method, 2341 is the return and if -5412 is the input to the method, -2145 is the return.
9. A *prime* number is defined as a number that is divisible by 1 and itself only. Write a Java program that asks the user to input an integer number *N*. It should then find the largest prime number less than *N* and the smallest prime number greater than *N*. In your program write and use a method called *prime* that takes an integer and returns 1 if the input is prime and returns 0, otherwise.
10. Write a method called `DigitAt()` that has two integer input parameters called **number** and **pos**. If **pos** is not positive or if it is greater than the number of digits in **number**, then the method returns -1, otherwise the method returns the digit in **number** at position **pos**. Some examples are listed below:

Input Parameters		Return Value	Explanation
<b>number</b>	<b>pos</b>		
342445	-8	-1	-8 is not positive
67335	9	-1	9 is greater than number of digits, which is 5
87452	1	2	The digit in location 1 is 2
8975302	4	7	The digit in location 4 is 7

11. A customer survey collects feedback regarding the food quality in a restaurant. Each score is between -5 (bad) to +5 (excellent). There are 10 scores to be input by the user.
- (a) Write a Java method named, `getFoodQuality()`, that prompts the user to input 10 scores of type integer. The method should calculate the average score and return the quality of the food based on the average score. If the average is less than -3.0, return "bad", if less than 0.0 "poor", if 0.0 "average", if less than or equal to 3.0 "quite good", if more than 3.0, "excellent". Note: If a particular score is outside the range -5 to +5, then the score is assumed to be spoiled, and a score of 0 is used instead.
- (b) Write a main method that calls the method defined in PART A and prints the quality of the food.

12. (A) Write a method called **Expansion** that accepts three parameters: **material** (string), **length** (float), and **temp\_change** (float). The method calculates the amount of object length change with respect to temperature change. Therefore, the method will return the **change in length** as follows:

$$\text{Change in length} = \text{coefficient} \times \text{length} \times \text{temp\_change}$$

where *coefficient* values are selected from the table (right) according to the **material**.

Material	Coefficient
Aluminum	24
Silver	18
other material	12

- (B) Write a complete program that prompts the user to enter Material (string), material length (float), and amount of change in temperature (float). The program should display the length change by calling the method defined in Part A. Format your output to **three** decimal points.
13. (A) Write a method named **calculateSUM** that takes 2 integers parameters START and END, where START is always less than or equal END. The method should return the sum of all integers between START and END (inclusive). For example, if the parameters values are START=5 and END=8 then the return value should be the sum of  $5 + 6 + 7 + 8 = 26$ .
- (B) Write a program that prompts the user to enter 2 integers representing START and END value where START is always less than or equal to END. The method should display the SUM of all integers between START and END by using the method defined in Part A. You need to define the method prototype before using the method. The Input/output of the program should be displayed as follows:

Sample input/Output 1

Enter START and END: 5 8

SUM is 26

Sample input/Output 2

Enter START and END: 9 9

SUM is 9

14. (A) Write a method called **DeliveryChange** that accepts two parameters: **carrier** (string) and **weight in kg** (int). The method will return the delivery cost according to the following:

$$\text{delivery cost} = \text{weight in kg} \times \text{cost per kg}$$

where *cost per kg* is selected from the following table according to the parameter **carrier**.

carrier	cost per kg
Car	3.0
Trailer	1.5
Drone	9.5

- (B) Complete the below program. First, prompt the user to enter delivery **carrier** (string) and **package weight** (int). Second, display the delivery cost by calling the method defined in Part A. Finally, call the method again to display standard delivery cost with delivery **carrier** “car” and package weight 1 kg. Follow the sample below and format your output to **three** decimal points.

Sample input/Output 1

Enter carrier and weight: Trailer 5  
The delivery cost is BD. 7.500  
Standard delivery cost is BD. 3.00

```
public class Question
{
    public static void main(String [] args)
    {
        ....
    }
}
```

15. (A) Write a method called **CustomsAmnt** that accepts three parameters: **Type** (String), **Quantity** (int), and **price** (double). The method will return the customs amount to be paid as follows:

$$\text{Customs amount} = \text{Handling fee} + \text{Tax} \times \text{Quantity} \times (\text{price} + \text{Tariff})$$

where *handling fee*, *Tax* and *Tariff* values are selected from the following table according to the parcel **Type**.

Type	Handling Fee	Tax	Tariff
DHL or FedEx	50	5%	10
other types	80	9%	30

- (B) Complete the below program to prompt the user to enter a parcel **type** (String), **quantity** of parcels (int) and parcel **price** (double). The program should display the customs amount by calling the method defined in Part (A). The input/output of the program should be displayed formatted to two decimal places as follows:

Sample input/Output 1

<i>Enter Type, Quantity and Parcel Price:</i> DHL    5        500
The customs amount is BD 203.00

```
public class Question
{
    public static void main(String [] args)
    {
        ....
    }
}
```

## 6 Classes & Objects

### 1. RetailItem Class

Write a class named **RetailItem** that holds data about an item in a retail store. The class should have the following fields:

- **description**. It is a **String** object that holds a brief description of the item.
- **unitsOnHand**. It is an **int** variable that holds the number of units currently in inventory.
- **price**. It is a **double** that holds the item's retail price.

Write appropriate mutator methods that store values in these fields and accessor methods that return the values in these fields. Once you have written the class, write a separate program that creates three **RetailItem** objects and stores the following data in them.

Description	Units On Hand	Price
Designer Jeans	40	34.95
Jacket	12	59.95
Shirt	20	24.95

### 2. Employee Class #1

Write a class named **Employee** that has the following fields:

- The **name** field is a **String** object that holds the employee's name.
- The **idNumber** is an **int** variable that holds the employee's ID number.
- The **department** field is a **String** object that holds the name of the department where the employee works.
- The **position** field is a **String** object that holds the employee's job title.

Write appropriate *set* methods that store values in these fields and *get* methods that return the values in these fields. Once you have written the class, write a separate program that creates three **Employee** objects to hold the following data.

Name	ID Number	Department	Position
Susan Meyers	47899	Accounting	Vice President
Mark Jones	39119	IT	Programmer
Joy Rogers	81774	Manufacturing	Engineer

### 3. Employee Class #2

Create a class called **Employee** that includes three instance variables — a first name (type **String**), a last name (type **String**) and a monthly salary (type **double**). Provide a constructor that initializes the three instance variables. Provide a *set* and a *get* method for each instance variable. If the monthly salary is not positive, set it to 0.0. Write a test app named **EmployeeTest** that demonstrates class **Employee**'s capabilities. Create two **Employee** objects and display each object's yearly salary. Then give each **Employee** a 10% raise and display each **Employee**'s yearly salary again.

### 4. Employee Class #3

**Part (1)** Define a class with the following specification.

- The class name is **Employee**. It has four instance variables: **firstName** (**String**), **payRate** (**double**), **normalHours** (**double**), and a **extraHours** (**double**).
- Make the instance variable **payRate** private and provide a mutator method (*set*) and an accessor method (*get*) to access this variable. In *set* method, if the supplied pay rate (parameter) is negative print an invalid message and do not change the instance variable **payRate**.
- A method **calcPay()** to calculate the employee pay by multiplying **payRate** by **normalHours** added to it the pay for **extraHours**. For every extra hour the pay rate is doubled. This method must return the pay as a result.
- A method **print()** that displays the normal and extra working hours and the calculated pay amount formatted to two decimal places as shown below.

#### Sample Output

Employee Name: Ahmed

Normal Hours Worked: 140.00

Extra Hours Worked: 20.00

Hourly Rate: BD 2.50

Pay Amount: BD 450.00

**Part (2)** Write a Java application that asks the user to enter the employee name, pay rate, normal hours worked, and extra hours worked. Create an object of type **Employee** class you defined in Part (1) and set its values according to the input data. Use it to calculate the pay amount and print the data.

### 5. Invoice Class

Create a class called **Invoice** that a hardware store might use to represent an invoice for an item sold at the store. An **Invoice** should include four pieces of information as instance variables — a part number (type **String**), a part description (type **String**), a quantity of the item being purchased (type **int**) and a price per item (type **double**). Your class should have a constructor that initializes the four instance variables. Provide a *set* and a *get* method for each instance variable. In addition, provide a method named *getInvoiceAmount* that calculates the invoice amount (i.e., multiplies the quantity by the price per item), then returns the amount as a **double** value. If the quantity is not positive, it should be set to 0. If the price per item is not positive, it should be set to 0.0. Write a test app named **InvoiceTest** that demonstrates class **Invoice**'s capabilities.

### 6. Car Class

Write a class named **Car** that has the following fields:

- The **yearModel** field is an **int** that holds the car's year model.
- The **make** field is a **String** object that holds the make of the car.
- The **speed** field is an **int** that holds the car's current speed.

In addition, the class should have the following methods.

- The constructor should accept the car's **year** model and **make** as arguments. These values should be assigned to the object's **yearModel** and **make** fields. The constructor should also assign 0 to the **speed** field.
- The appropriate accessor methods get the values stored in an object's **yearModel**, **make**, and **speed** fields.
- The *accelerate* method should add 5 to the **speed** field each time it is called.
- The *brake* method should subtract 5 from the **speed** field each time it is called.

Demonstrate the class in a program that creates a **Car** object, and then calls the *accelerate* method five times. After each call to the *accelerate* method, get the current speed of the car and display it. Then, call the *brake* method five times. After each call to the *brake* method, get the current speed of the car and display it.

### 7. Temperature Class

Write a Temperature class that will hold a temperature in Fahrenheit and provide methods to get the temperature in Fahrenheit, Celsius, and Kelvin. The class should have the following field:

- **ftemp** — A **double** that holds a Fahrenheit temperature.

The class should have the following methods:

- The constructor accepts a Fahrenheit temperature (as a **double**) and stores it in the **ftemp** field.
- *setFahrenheit* — The method accepts a Fahrenheit temperature (as a **double**) and stores it in the **ftemp** field.
- *getFahrenheit* — Returns the value of the **ftemp** field, as a Fahrenheit temperature (no conversion required).
- *getCelsius* — Returns the value of the **ftemp** field converted to Celsius.
- *getKelvin* — Returns the value of the **ftemp** field converted to Kelvin.

Use the following formula to convert the Fahrenheit temperature to Celsius:

$$Celsius = (5/9) \times (Fahrenheit - 32)$$

Use the following formula to convert the Fahrenheit temperature to Kelvin:

$$Kelvin = ((5/9) \times (Fahrenheit - 32)) + 273$$

Demonstrate the **Temperature** class by writing a separate program that asks the user for a Fahrenheit temperature. The program should create an instance of the **Temperature** class, with the value entered by the user passed to the constructor. The program should then call the object's methods to display the temperature in Celsius and Kelvin.



## 8. The Stock Class

Design a class named **Stock** that contains:

- A string data field named **symbol** for the stock's symbol.
- A string data field named **name** for the stock's name.
- A double data field named **previousClosingPrice** that stores the stock price for the previous day.
- A double data field named **currentPrice** that stores the stock price for the current time.
- A constructor that creates a stock with the specified symbol and name.
- A method named *getChangePercent()* that returns the percentage changed from **previousClosingPrice** to **currentPrice**.

Draw the UML diagram for the class and then implement the class. Write a test program that creates a **Stock** object with the stock symbol *ORCL*, the name *Oracle Corporation*, and the previous closing price of 34.5. Set a new current price to 34.35 and display the price-change percentage.

## 9. The Room Carpet Class

**Part (1)** Create a class called **RoomCarpet** that has three private instance variables: room length and width in foot (**double**) and carpet cost per square foot (**double**). Additionally, the class should have:

- (a) A constructor that takes room length, room width and carpet cost per square foot. The constructor sets the class variables to these values and in case any of these values is negative, set its instance variable to zero.
- (b) A mutator method (*set*) and an accessor method (*get*) for the carpet cost variable *only*.
- (c) A method **getArea** that returns the area of the room. The area of the room is computed by multiplying the room's length by the room's width.
- (d) A method **getTotalCost** that returns the total cost of the carpet. The total cost is computed by multiplying the room area by the carpet cost per square foot.

**Part (2)** Write a Java application that asks the user to enter the length and width of a room and the cost per square foot of the carpeting. Create an object from **RoomCarpet** class you defined in Part (1) and use it to display the room area and the total cost of the carpet.

## 10. Spider Class

**Part (1)** Define a class with the following specification.

- (a) The class name is **Spider** and has four instance variables: **name** (String), **poison** (boolean), **weight** (double), and **gender** (char).
- (b) Provide *set* and *get* methods to access the **gender** variable. In *set* method, the passed parameter should be **M** or **m** for male and **F** or **f** for female. If not, your program should print an invalid message and set **gender** to **V** as invalid gender.
- (c) Write a method called **determineEnvironment()** that accepts no input parameter and returns a string value for the environment. If the weight is in the range of 120 to 170 and it is poisonous, the method returns "jungles" as the environment, if the weight is in the range of 120 to 170 and it is not poisonous, the method returns "desert" as the environment, and any other cases it returns "Jungles/desert" as the environment.
- (d) Write an **equals()** method for this class. The method should accept a **Spider** object as an argument. It should return true if the argument object contains the same name as the calling object or false otherwise.
- (e) Write a method called **print()** that displays all the below data.

### Sample Output

Spider Name: Black Widow

Poisonous: Yes

Weight: 120.0

Gender: Female

Environment: Jungles

**Part (2)** Write a Java application to do the following:

- (a) Ask the user to enter the name, weight, and gender for a poisonous spider.
- (b) Create an object called **spider1** from the class **Spider** and store the data entered by the user to its values.

- (c) If `spider1` name is “Black Widow”, then print all the spider data. Otherwise, print “It is not poisonous”.

### 11. Bank Account Class

**Part (1)** Define a class with the following specification.

- (a) The class name is `BankAccount` and it has two private variables: `fullName` (String) and `balance` (double).
- (b) Write a constructor that initializes `fullName` and `balance` variables. If `balance` is negative, then set it to 0.0 and print the message “Invalid balance. Balance is set to 0.0”.
- (c) Provide a `set` and `get` methods to `balance` variable. In `set` method, if the supplied `balance` (parameter) is negative print an invalid message and do not change the `balance`.
- (d) Write a method `deposit()` that has one double input parameter called `amount`. The method adds `amount` to the current `balance`.
- (e) Write a method `withdraw()` that has one double input parameter called `amount`. The method subtracts `amount` from the current `balance`.
- (f) Write method `printlnInfo()` that displays the full name on a single line followed by the `balance` on the next line.

#### Sample Output

Full Name: Yasser Abbas

Balance: 1240.250

**Part (2)** Part (2) Write a Java application to test the class you defined in Part (1):

- (a) Ask the user to enter the full name followed by the initial account balance.
- (b) Create an object of type `BankAccount` class and initialize it by the user data using its constructor.
- (c) Deposit BD 135 to the account followed by withdraw BD 35 from the account.
- (d) Print the account name and balance.

### 12. Football Team Class

**Part (1)** Define a class with the following specification.

- (a) The class name is `FootballTeam`. It has three private instance variables: `tname` (String), `wins` (int) and `losses` (int).
- (b) Write a constructor that accepts the name of the team, the number of wins, and the number of losses as arguments and sets the class properties to those values. This constructor should be public and if `wins` or `losses` parameters are negative, then the constructor sets them to 0.
- (c) Write public getter methods for all the instance variables.
- (d) Write a method called `win()` to increase the number of wins by one and another method called `lose()` to increase the number of losses by one.
- (e) Write a public method called `hasGoodRecord()` that returns true when a team has a “good record,” meaning the team has more wins than losses or returns false otherwise.
- (f) Write an `equals()` method for this class. The method should accept a `FootballTeam` object as an argument. It should return true if the argument object contains the same name as the calling object or false otherwise.
- (g) Write a method called `printlnInfo()` that displays all the below data.

#### Sample Output

Team Name: Zallaq

Wins: 9

Losses: 4

**Part (2)** Write a Java application to do the following:

- (a) Ask the user to enter the name, the number of wins, and the number of losses for a team.
- (b) Create an object called `myTeam` from the class `FootballTeam` and initialize its values to the user data.
- (c) If `myTeam` has good record, then print its info data, otherwise, find and print the number of wins needed to make the team have good record.

### 13. The Course Class

**Part (1)** Define a class with the following specification.

- (a) The class name is **Course**. It has three private instance variables **code** (String), **credits** (int) and **grade** (char).
- (b) Write a constructor that accepts the code, number of credits and a grade of a course. If credit parameter is negative, set **credits** to zero and if grade parameter is not A, B, C, or D, set **grade** to F. Assume that all grades are in uppercase.
- (c) Write a public *getter* and *setter* for **credits** variable. In the setter method, keep **credits** unchanged in case the input parameter is negative.
- (d) Write a public method called **weight()** that returns the course weight. The course weight is computed by multiplying the course credits with grade points (i.e.  $weight = course\ credits \times grade\ points$ ) according to the table below:

Grade	A	B	C	D	F
Points	4.0	3.0	2.0	1.0	0.0

- (e) Write a public method called **printInfo()** that displays all the instance variables. An example is shown below:

**Sample Output**

Code: ITCS113

Credits: 3

Grade: B

**Part (2)** Write a Java application to do the following:

- (a) Create two objects from the class **Course**. One for ITCS113 with number of credits is 5 and grade is A, and the other is for PHYSC101 with number of credits is 2 and grade is C.
- (b) Change the credit hours for ITCS113 to 3 and PHYSC101 to 4.
- (c) Print the course info for both courses.
- (d) Compute and print the GPA for the two courses using the equation:

$$GPA = \frac{Course\ 1\ weight + Course\ 2\ weight}{Course\ 1\ credits + Course\ 2\ credits}$$

#### 14. Building Class

**Part (1)** Define a class with the following specification.

- (a) Create a class called **Building** that has three private data members: **name** (String), number of **flats** (int) and **roomsPerFlat** (int).
- (b) Write a constructor that accepts only the name of the building as an input parameter. The constructor sets the data member **name** to the input parameter and initializes **flats** and **roomsPerFlat** to 0.
- (c) Provide only one accessor (get) method for the **name** data member.
- (d) Provide a method **setBuildingDetails** that sets both **flats** and **roomsPerFoot** data members from input parameters. Any of the input parameters is negative leave its corresponding data member unchanged.
- (e) Define a method called **calculateCapacity()** which calculates and returns the capacity (double) of the building. The capacity is calculated using the formula:

$$Capacity = \frac{flats \times roomsPerFlat}{factor}$$

where *factor* is equal to 0.75.

- (f) Define a method called **displayBuildingDetails()** to display on the screen all the building's details as in the following sample:

**Sample Output**

Name: Andalus

Flat: 20, Rooms: 3, Capacity: 80

**Part (2)** Write a Java application to do the following:

- (a) Create an object called **b1** from the class **Building** and initialize its name to "Awal".
- (b) Set the number of flats of **b1** to 12 and the number of rooms per flat to 3.
- (c) Display the building details for **b1**.

15. **Marathon Runner Class**

**Part (1)** Define a class with the following specification.

- Create a class called **MarathonRunner** that has three private data members: **name** (String), **distance** (double) and **time** (double).
- Write a constructor that accepts **name**, **distance**, and **time** as input parameters to initialize the private data members.
- Provide only one accessor (get) method and one mutator (set) method for **distance** data member.
- Define a private method called **calculateSpeed()** which calculates and returns the speed (double) of the marathon runner. The speed is calculated using the formula:

$$speed = \frac{distance}{time}$$

- Define a public method called **isFaster()** that accepts as input parameter a speed record(double). The method should return true if the current runner's speed is faster than the speed record, otherwise, return false.
- Define a public method called **displayDetails()** to display on the screen the runner's details as in the following sample:

**Sample Output**

Runner name: Ahmed  
 Distance traveled: 11700 m  
 Time elapsed: 2372.0 s  
 Runner's Speed: 4.93 m/s

**Part (2)** Write a Java application to do the following:

- Asks the user to enter a runner's first name, distance traveled and time elapsed. Create an object called **runner1** from class **MarathonRunner** you defined in *previous question* and set its values according to the input data.
- Display the details of **runner1**.
- Display the message "You broke the speed record" if **runner1** is faster than the speed record of value 6.7 (m/s).

16. **Event Class**

**Part (1)** Define a class with the following specification.

- Create a class called **Event** for a conference. The class has three private fields: event **code** (String), number of **guests** for the event (int), and the **price** (double). The event code is stored as a String such as *M312*.
- Write a public set method that sets the event code.
- Write a public set method that sets the number of guests. In this method, it will set the number of guests as well as compute the price (no set method for the price data member). The price is computed by multiplying the number of guests by a fixed price of BD. 10. If the number of guests is greater than 50, then the price is discounted by 10%.
- Write a public get method that returns the price value.
- Write a constructor that initializes the event code and number of guests. Set the number of guests inside the constructor by calling its set method defined in part (c).
- Write a public method called **isCrowded()** that returns a boolean value. The method returns true if the number of guests is greater than 100.
- Write a method to print the event code, number of guests, and the price.

**Part (2)** Write a Java application to do the following:

- Ask the user to enter the event code and the number of guests.
- Create an object called **myEvent** from the class *Event* and initialize its values to the user data.
- If **myEvent** is crowded, then set the number of guests to 100 and print a message to the user about this change.

- (d) Print **myEvent** information.

#### Sample Output

```
Enter event's code and number of guests:  M432 113
Event Crowded, the number of guests has been set to 100
Event code: M432
Number of Guests: 100
Price: 900.0
```

### 17. Height Weight Ratio Class

**Part (1)** Define a class with the following specification.

- Create a class called **HeightWeightRatio** with three private data members: **gender** (String as male or female in any letter case), **height** (double in cm), and **weight** (double in kg).
- Write a constructor that accepts **gender**, **height** and **weight** as input parameters to initialize the private data members.
- Provide an accessor (**get**) method and a mutator (**set**) method for the **height** data member.
- Define a **public method** called **checkRatio()** which checks the height/weight ratio and returns the **decision** as Boolean. The height/weight ratio decision is checked as the following:
  - Male: Acceptable when **weight** is between (height-105) and (height-95); otherwise, is not acceptable.
  - Female: Acceptable when **weight** is between (height-100) and (height-90); otherwise, is not acceptable.
- Define a **public method** called **showDetails()** to show all details as in the following sample:

Gender is Male Height is 170 CM Weight is 74 KG
---

**Part (2)** Write a Java application to do the following:

- Asks the user to input his/her gender, height and weight.
- Create an object called **person1** from the class **HeightWeightRatio** and initialize its private members according to the input data.
- Display the details of **person1** object.
- Display the message: “Acceptable Hight/Weight Ratio” OR “Not Acceptable Hight/Weight Ratio” based on the **checkRatio()** decision.

#### Sample Output

```
Input Gender, Height and Weight: Male 170 74
Gender is Male
Height is 170 CM
Weight is 74 KG
Acceptable Hight/Weight Ratio.
```

### 18. Car Class

**Part (1)** Define a class with the following specification.

- Create a class called **Car** that has three private data members: **name** (String), **torque** (double) in foot pounds and **RPM** (int) of the car engine.
- Write a constructor that accepts **name**, **torque** and **RPM** as input parameters to initialize the private data members. Assume all input parameters values are valid inputs.
- Provide only one accessor (**get**) method and one mutator (**set**) method for the **torque** data member.
- Define a **private method** called **calculateHorsePower()** which calculates and returns the horsepower (double) of the car. The horsepower is calculated using the formula:  $horsepower = torque \times RPM / 5252$ .
- Define a **public method** called **hasMoreHP()** that accept as input parameter an object of type **Car**, holding the details of another car. The method should return true if the current car's horsepower is bigger than the other car's horsepower, otherwise, false should be returned.
- Define a **public method** called **displayDetails()** to display on the screen the runner's details as in the following sample:

Car name: BMW750M3  
 Torque: 2250.95  
 Engine's RPM: 999  
 Horsepower: 428.1605198019802

**Part (2)** Write a Java application to do the following:

- (a) Asks the user to enter a car's name, torque and engine's RPM. Create an object called **car1** from class **Car** you defined in **Part (1)** and set its values according to the input data.
- (b) Create another object called **car2** from the class **Car** and initialize its name to "LexusLS600", its torque to 2050.5 and its RPM to 950.
- (c) Display only the car details who has bigger horsepower on the Screen according to their torque and RPM.

#### 19. Wind Speed Class

Wind speed often is measured in kilometer per hours (km/h) or knots (nautic miles). Another measure is the Beaufort scale consisting of integer numbers from 1 to 12. Design and implement a Java class, for creating wind speeds. Implement a constructor to create wind speed for a given velocity in kilometers per hour. The class must contain methods for getting the speed as knots or on the Beaufort scale. Furthermore, it must be possible to check whether the wind is calm or a European windstorm (Orcan). Wind speeds less than 2 km/h are calm. More than 120 km/h is a European windstorm. A nautic mile is 1,852 Kilometer. The Beaufort scale is defined by  $v = 3,01 * B^{3/2}$ , with  $v$  as the wind velocity in km/h. The Beaufort value  $B$  is rounded to the nearest integer. There are no Beaufort values larger than 12.

**Hint:** You can calculate  $a^b$  with `Math.pow(a,b)`.

#### 20. Rational Numbers Class

Rational numbers are numbers that can be represented as a fraction  $p/q$  where  $p$  is an integer number and  $q$  is a positive integer ( $q \neq 0$ ). Design and implement a Java class *RationalNumber* for representing such numbers. Implement methods to add and multiply rational numbers. Implement a method for return the value of a rational number as a double value. Make sure that the numerator  $p$  and denominator  $q$  do not have common divisors in your implementation. Use the algorithm for calculation the greatest common divisor to ensure this property. Mind, that zero has a unique internal representation in your implementation.

#### 21. Payroll Class

Design a Payroll class that has fields for an employee's name, ID number, hourly pay rate, and number of hours worked. Write the appropriate accessor and mutator methods and a constructor that accepts the employee's name and ID number as arguments. The class should also have a method that returns the employee's gross pay, which is calculated as the number of hours worked multiplied by the hourly pay rate. Write a program that demonstrates the class by creating a **Payroll** object, then asking the user to enter the data for an employee. The program should display the amount of gross pay earned.

#### 22. Test Scores Class

Design a TestScores class that has fields to hold three test scores. The class should have accessor and mutator methods for the test score fields and a method that returns the average of the test scores. Demonstrate the class by writing a separate program that creates an instance of the class. The program should ask the user to enter three test scores, which are stored in the **TestScores** object. Then the program should display the average of the scores, as reported by the **TestScores** object.

#### 23. Swimming Pool Class

Write the definition of a class, **swimmingPool**, to implement the properties of a swimming pool. Your class should have the instance variables to store the length (in feet), width (in feet), depth (in feet), the rate (in gallons per minute) at which the water is filling the pool, and the rate (in gallons per minute) at which the water is draining from the pool. Add appropriate constructors to initialize the instance variables. Also add member functions, to do the following: Determine the amount of water needed to fill an empty or partially filled pool; the time needed to completely or partially fill the pool, or empty the pool; add water or drain for a specific amount of time.

#### 24. Line Class

The equation of a line in standard form is  $ax + by = c$ , where  $a$  and  $b$  both cannot be zero, and  $a, b$ , and  $c$  are real numbers. If  $b \neq 0$ , then  $-a/b$  is the slope of the line. If  $a = 0$ , then it is a horizontal line, and if  $b = 0$ , then it is a vertical line. The slope of a vertical line is undefined. Two lines are parallel if they have the same slope or both are vertical lines. Two lines are perpendicular if one of the lines is horizontal and another is vertical, or if the product of their slopes is  $-1$ . Design the class Line to store a line. To store a line, you need to store the values of  $a$  (coefficient of  $x$ ),  $b$  (coefficient of  $y$ ), and  $c$ . Your class must contain the following operations:

- (a) If a line is nonvertical, then determine its slope.
- (b) Determine if two lines are equal. (Two lines  $a_1x + b_1y = c_1$  and  $a_2x + b_2y = c_2$  are equal if either  $a_1 = a_2, b_1 = b_2$ , and  $c_1 = c_2$  or  $a_1 = ka_2, b_1 = kb_2$ , and  $c_1 = kc_2$  for some real number  $k$ .)
- (c) Determine if two lines are parallel.
- (d) Determine if two lines are perpendicular.
- (e) If two lines are not parallel, then find the point of intersection.

Add appropriate constructors to initialize variables of Line. Also write a program to test your class.