INTRODUCTION TO JAVA LANGUAGE

Java was created at Sun Microsystems, Inc., where James Gosling led a team of researchers in an effort to create a new language that would allow consumer electronic devices to communicate with each other. Java was first released in 1995, and Java's ability to provide interactivity and multimedia showed that it was particularly well suited for the Web.

- **Java**, is a modern object-oriented computer programming language. Object-oriented programming (OOP) is a programming technique **that uses objects** to design the **applications and computer programs. In Java, Class** is a template or blueprint to create an object. Class is a collection of objects of similar type. Class must be defined before creating an object. Classes are generic. We can create objects from each class.
- A **class** is codes that define the behavior of a Java Programming element called an object. It is a description of the properties and the behaviors of one or more objects. It defines the variable and functions that are common to the objects of its type.
- **An object** is an entity that has both properties/ state and behaviour. The states/ properties of an object consists of any data that the object might be keeping track of. The behaviour consists of actions that the object can perform. Object does things and usually depends on its properties. Objects are specific. Objects are created from classes.

EXAMPLES OF CLASS-OBJECT

- Consider the following example to understand the class-object concept clearly.
- A car is a real world object and it has its own characteristics like colour, size, model, engine capacity and so on.
- These are data members of the car.
- You can drive the car, take a turn using the car or stop the car.
- These are the member functions or the methods.
- Figure 3 represents the car object with its data member and member functions.

Figure 3 : Car Object

- Figure 4 explains the Object-Oriented Approach using the vehicle class example.
- Hence, a vehicle is a class that defines the data members and member functions that are common to the objects, such as a car, a bus and a truck.

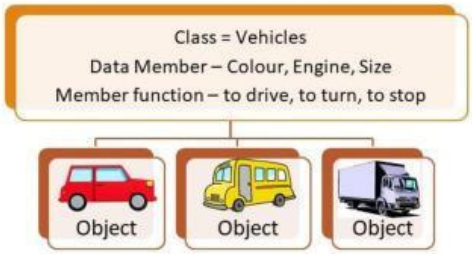


Figure 4 : Basic Elements of Object-Oriented Approach

Basic java code that student must know.

INSTRUCTION	JAVA CODE

Create class java program name HelloWorld	class HelloWorld
	{
	
	}
	//Open and close braces, { } indicate the start and end of the class.

Create method main( ) that compulsory in Java class to run the program or call method	public static void main (String [ ] args)
	{
	
	}
	//Open and close braces, { } indicate the start and end of the class.

Create output statement to display message "Hello, world" on the console	System.out.print("Hello, world!");
Create output statement to display the contents of variables age	System.out.println("Age: "+ age);

Declare variables based on suitable data type	String name="Murni";
	int age=18;
	double weight=65.4;
	boolean married=true;
	char gender='F';

Allow Java to read input from keybord, which allows user to enter some input and it will parse into data types have 3 step 1. Import Scanner class which is define in java.util package 2. Creat an object of Scanner class 3. Create input statement and assign the value to a variable named. 4. You also can create user prompt	1. import java.util.Scanner; 2. Scanner sc = new Scanner(System.in); 3. int num1=sc.nextInt(); 4. System.out.print("Enter two numbers:

Full java program that using input statement and output statement	import java.util.Scanner;
	class ArithmeticMain
	{
	public static void main (String [ ] args)
	{
	Scanner sc = new Scanner(System.in);
	System.out.print("Enter two numbers: ");
	int num1=sc.nextInt();
	int num2=sc.nextInt();
	}
	}

Notes: Sequence Control Structure

Case Study 1: To make a profit, the prices of the items sold in a furniture store are marked up by 60%. Calculate the selling price of an item sold at the furniture store.

Step 1: Problem Analysis

Input	Process	Output
item price	Calculate the selling price of an item based on the item price entered by the user.	selling price

Step 2: Design a Solution

Pseudocode

Start

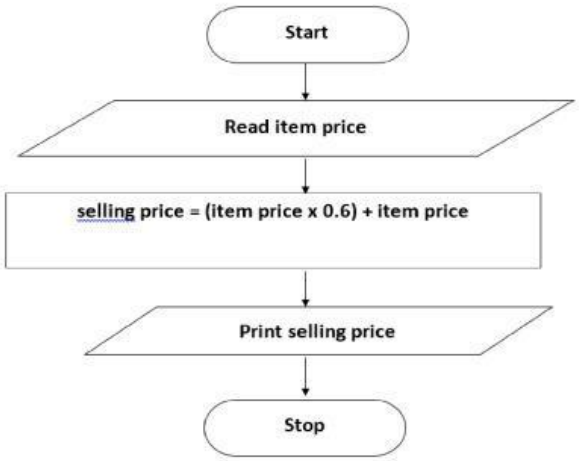
Read item price

selling price = (item price x 0.6) + item price

Print selling price

Stop

Flowchart



Step 2: Design a Solution (Pseudocode)	Step 3: Implementation (Java)
Start	public static void main (String [ ] args)
	{
	Scanner sc = new Scanner (System.in);
	double itemprice, sellingprice;
Read item price	*System.out.print("Enter price : RM ");* *//may with or without user friendly statement.* itemprice = sc.nextDouble();
selling price = (item price x 0.6) + item price	sellingprice = (itemprice*0.6) + itemprice;
Print selling price	System.out.print ("The selling price of an item sold at the furniture store is: RM " + sellingprice);
Stop	}

Step 3: Implementation

```java
//*Import Statements
import java.util.Scanner;

//1.Class Header
public class Case3Celcius

//2.Start of Class Body
{
    //3.Main Method Header
    public static void main(String[] args)

    //4.Start of Method Body
    {
        //Creating Objects
        Scanner sc = new Scanner(System.in);

        //Declaring variables
        double itemprice,sellingprice;

        System.out.print("Please enter the the item price of the furniture: RM  ");
        //Read Input
        itemprice = sc.nextDouble();

            //Process Statements
            sellingprice = (itemprice*0.6) + itemprice;

        //Print Output
        System.out.print ("The selling price of an item sold at the furniture store is: RM " +  sellingprice);

    }//5.End of Method Body

}//6.End of Class Body
```

Step 4: Testing

```
Bluej: Terminal Window - Sequence
Options
Please enter the the item price of the furniture: RM  850
The selling price of an item sold at the furniture store is: RM 1360.0
```

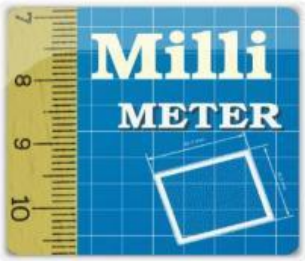
Drag and Drop Exercise

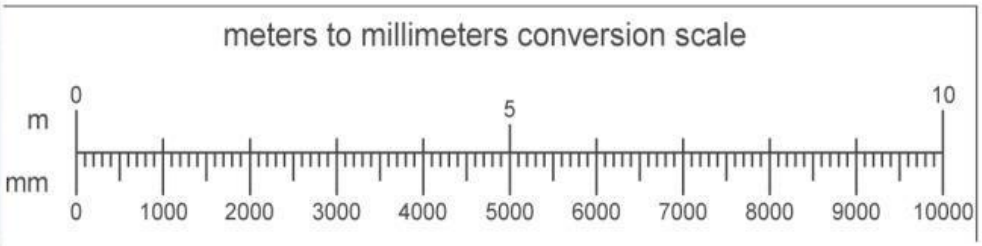
Name: _____

Class: _____

Matching algorithm with Java Code

Instruction : Identify and match with the right Java code.





Pseudocode	Java code
Start	public static void main (String [ ]args)
	{
	Scanner sc = new Scanner (System.in);
Read milimeter	
meter = milimeter / 1000	
Print meter	
Stop	}

Drag and drop your answer.

;

meter = milimeter / 1000;

milimeter = sc.nextDouble( )

System.out.print("Enter milimeter value: ");

double milimeter, meter;

System.out.print("The value in meter is " + meter);

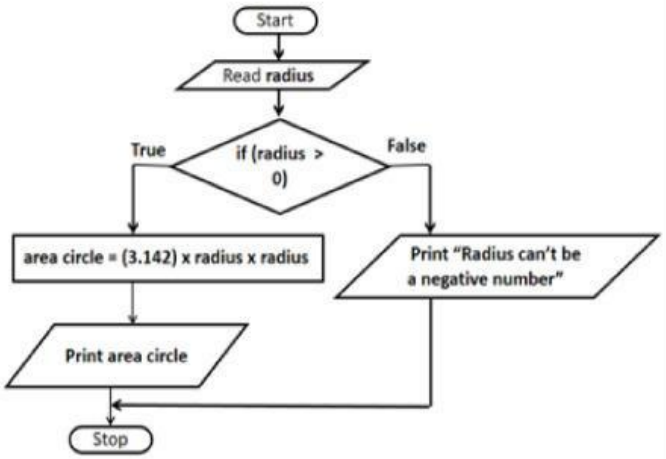
Notes : Selection Control Structure

Case Study 2: Calculate the area of a circle if the radius entered is a positive value, else print the message "Radius can't be a negative number".

Step 1: Problem Analysis

Input	Process	Output
radius	***Determine and calculate*** area of a circle ***based on*** the radius *entered by user.*	*Either* area circle or the message "Radius can't be a negative number"

Step 2: Design a Solution (Pseudocode)	Step 2: Design a Solution (Flowchart)

Start
 	
Read radius
if (radius > 0)
area circe = (3.142) x radius x radius
Print area circle
else
Print "Radius can't be a negative number"
end if
 	
Stop
	

Step 2: Design a Solution (Pseudocode)	Step 3: Implementation
Start	{
	double radius, areacircle;
Read radius	radius=sc.nextDouble( );
if (radius > 0)	***if*** (radius > 0)
area circe = (3.142) x radius x radius	{ areacircle = (3.142) *radius * radius;
Print area circle	System.out.print (areacircle);}
else	***else***
Print "Radius can't be a negative number"	System.out.print("Radius can't be a negative number");
end if	***//end if***
Stop	

Don't Do It...
No semicolon at the end of the ***if*** statement because the statement does not end here.

Good Programming Practice...
Although not a requirement, it is a good programming practice to use a comment (//end if) to mark the end of the ***if*** statement in a program.

Remember... Syntax-if statement in JAVA

```
if (condition)
  {
     multiple actions/statements
  }
//end if
```

Step 3: Implementation

```java
//*Import Statements
import java.util.Scanner;
//1.Class Header
public class C3IfElse
{//2.Start of Class Header

    //3.Main Method Header
    public static void main(String[] args)
    {//4.Start of Method Body

            //Creating Objects
            Scanner sc = new Scanner(System.in);

            //Declaring variables
            double radius,areacircle;

            System.out.print("Please enter a radius ");
            //Read input
            radius = sc.nextDouble();

                //Process
                if (radius > 0)
                    {areacircle = (3.142)*(radius*radius);
                    //Print Output
                    System.out.print ("The area of circle is  "+areacircle);}
                else
                    //Print Output
                    System.out.print("Radius can't be a negative number");
                    //end if
    }//5.End of Method Body
}//6.End of Class Body
```

Step 4: Testing

```
BlueJ: Terminal Window - Selection
Options
Please enter a radius 4.5
The area of circle is  63.625499999999995
```

```
BlueJ: Terminal Window - Selection
Options
Please enter a radius -9
Radius can't be a negative number
```

Drag and Drop Exercise

Name: _____

Class: _____

Matching algorithm with Java Code

Instruction : Identify and match with the right Java code

PSEUDOCODE	JAVA CODE
Start	public static void main (String [ ]args) { Scanner sc = new Scanner (System.in);
Read amount paid	
if (amount paid > RM500)	
	{
total discount = 0.15 x amount paid	
amount after discount = amount paid x 0.85	
	}
else	
total discount = 0.1 x amount paid	
amount after discount = amount paid x 0.90	
end if	
Display total discount, amount after discount	
Stop	}

Drag and drop your answer.

System.out.println("The amount after discount is RM: " + amountafterdiscount);

}

{

else

;

System.out.println("The total discount is RM: " + totaldiscount);

System.out.print("Enter amount paid : RM ");

double amountpaid, totaldiscount, amountafterdiscount;

totaldiscount = 0.15 x amountpaid;

amount after discount = amount paid x 0.90

totaldiscount = 0.1 x amountpaid;

amountafterdiscount = amountpaid x 0.85;

amountpaid = sc.nextDouble( );

if (amountpaid > 500.00)

Notes : Repetition Control Structure (Counter Controlled)

Case Study 3: To make a profit, the prices of the items sold in a furniture store are marked up by 60%. Calculate the selling price of 6 items sold at the furniture store.

Step 1 : Problem Analysis

Input	Process	Output
item price	Calculate the selling price based on the item price entered by the user and repeat the program for 6 times.	selling price

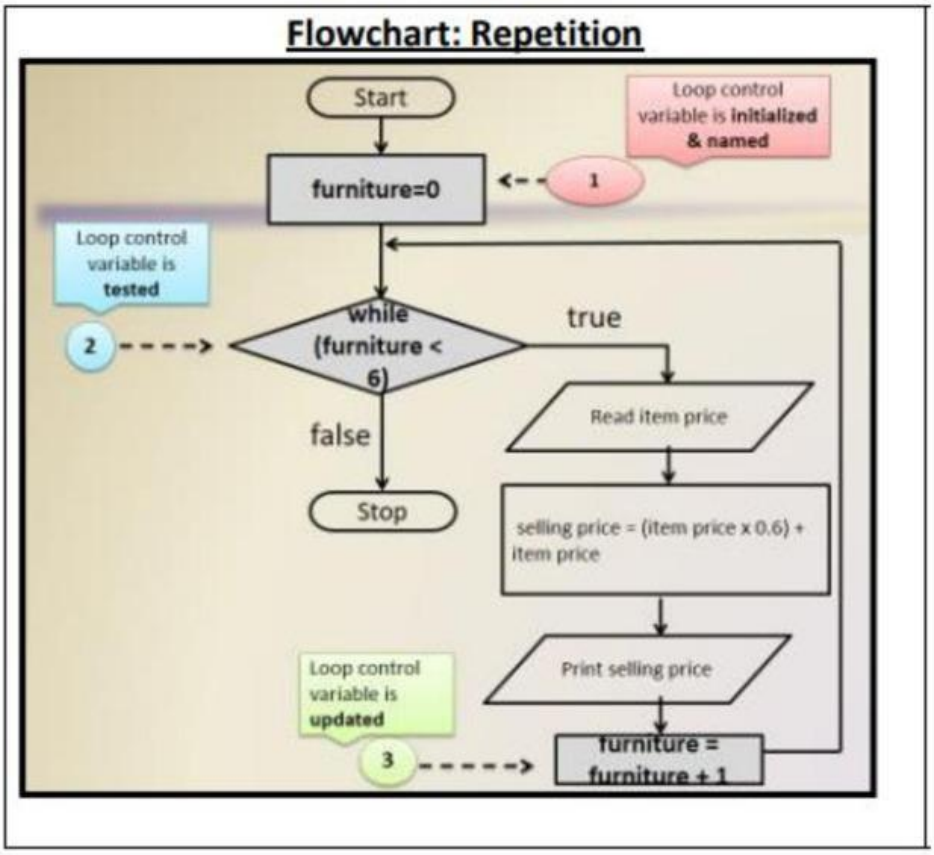
Step 2 : Design a solution

Pseudocode

Pseudocode Repetition

```
Start
        furniture = 0
        repeat while ( furniture < 6)
            Read item price
            selling price = (item price x 0.6) + item price
            Print selling price
        furniture = furniture + 1
        end repeat
Stop
```

Flowchart



Step 2 : Design a solution	Step 3: Implementation
Start	public static void main(String[] args)
	{
	Scanner sc = new Scanner(System.in);
Set furniture = 0;	int furniture = 0;
	double itemPrice, sellingPrice;
repeat while (furniture < 6)	while(furniture < 6)
	{
Read item price	System.out.print("Enter price of item : RM "); //for user friendly program
	itemPrice = sc.nextDouble();
selling price = (item price x 0.6) + item price	sellingPrice = (itemPrice * 0.6) + itemPrice;
Print sellingPrice	System.out.println("Selling price : RM"+sellingPrice);
furniture = furniture + 1	furniture = furniture + 1;
End repeat	}
Stop	} //end of main method

Exercise

Name: _____

Class: _____

Matching algorithm with Java Code
Instruction : Identify and match with the right Java code.

Create a program that calculates and displays the pay of 5 employees, given hours worked and hourly pay rate.

Pseudocode	Java statement
Start	public static void main(String[ ] args) { Scanner sc = new Scanner(System.in);
Set employee = 1	
while(employee ≤ 5)	
Read hours worked, hourly pay rate	
Pay = hours worked x hourly pay rate	
Print Pay	
employee = employee + 1	
End while	}
Stop	}

{ ; while (employee <= 5) employee = employee + 1; hoursWorked = sc.nextDouble( );

pay = hoursWorked * hourlyPayRate int employee = 1; System.out.println("Pay : RM"+pay);

double hoursWorked, hourlyPayRate, pay; hourlyPayRate = sc.nextDouble( );

System.out.print("Read hours worked and hourly pay rate : ");