PRACTICAL MANUAL

Agri-Informatics (Agricultural Informatics)

BSc Agriculture III Semester ABB 252 2(1+1) &
BSc Forestry VIII Semester FBS 442 3(2+1)

Dr. Amit Kumar Jain
Dr. Tanuj Misra

2020

College of Agriculture
Rani Lakshmi Bai Central Agriculture University,
Jhansi - 284003

Syllabus:

Study of Computer Components, accessories, practice of important DOS Commands. Introduction of different operating systems such as windows, Unix, Linux, Creating, Files & Folders, File Management. Use of MS-WORD and MS Power point for creating, editing and presenting a scientific Document, Handling of Tabular data, animation, video tools, art tool, graphics, template & designs. MS-EXCEL - Creating a spreadsheet, use of statistical tools, writing expressions, creating graphs, analysis of scientific data, handling macros. MS-ACCESS: Creating Database, preparing queries and reports, demonstration of Agri-information system. Introduction to World Wide Web (WWW) and its components, creation of scientific website, presentation and management agricultural information through web. Introduction of various programming languages such as Visual Basic, Java, Fortran, C, C++, and their components Hands on practice on writing small programmes. Hands on practice on Crop Simulation Models (CSM), DSSAT/Crop-Info/CropSyst/ Wofost. Preparation of Inputs file for CSM and study of model outputs, computation of water and nutrient requirements of crop using CSM and IT tools. Use of smart phones and other devices in agro-advisory and dissemination of market information. Introduction of Geospatial Technology, demonstration of generating information important for Agriculture. Hands on practice on preparation of Decision Support System.

Name of Students .....

Roll No. .....

Batch .....

Session ..... Semester: .....

Course Name : .....

Course No. : .....

Credit .....

Published: 2020
No. of copies: ……..
Price: Rs.

CERTIFICATE

This is to certify that Shri./Km. .....ID No..... has completed the practical of course..... course No. ..... as per the syllabus of B.Sc. (Hons.) Agriculture/ Horticulture/ Forestry ..... semester in the year.....in the respective lab/field of College.

Date: Course Teacher

INDEX

Practical No. 1

Objective: To know about Central Processing Unit

Define:

Central Processing Unit:

……..

……

……

……

……

Process:

……

……

……

Memory Unit: ……..

……

……

Primary Memory: ……..

……

……

……

……...

……

……...

……

……...

……

……

……

Secondary Memory: ……..

……

……...

……

……...

……

……

……

Control unit: ……..

……

……

……

……

Exercise:

Q.2 Write the main work of ALU?

……

……

……

……

……

……

……

……

Q.3 Write down primary work of RAM & ROM.?

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

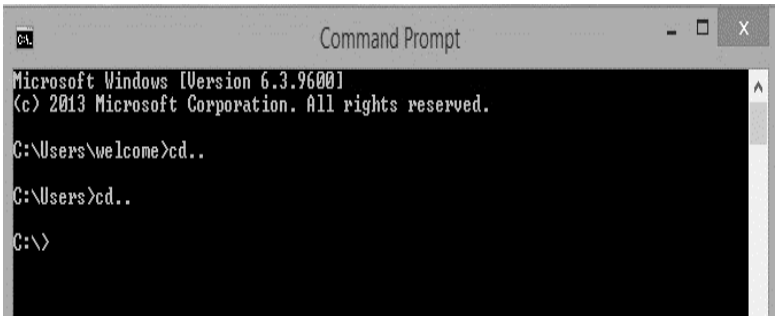
……

……

Practical No 2

Objective: To learn about DOS Commands

DOS (Disk Operating System) …



……

……

……

……

……

……

……

……

Command	Description	Type
ansi.sys		
append		
arp		
assign		
assoc		
at		
atmadm		
attrib		
batch		
bcdedit		
break		
cacls		
call		
cd		
chcp		
chdir		
chkdsk		
chkntfs		

choice		
clip		
cls		
cmd		
color		
command		
comp		
compact		
control		
convert		
copy		
ctty		
date		
debug		
defrag		
del		
delete		
deltree		
dir		
diskcomp		
diskcopy		
doskey		
dosshell		
driverquery		
drivparm		
echo		
edit		
edlin		
emm386		
endlocal		
erase		
exit		

expand		
extract		
fasthelp		
fc		
fdisk		
find		
findstr		
for		
format		
ftp		
fType		
goto		
graftabl		
help		
if		
ifshlp.sys		
ipconfig		
keyb		
label		
lh		
listsvc		
loadfix		
loadhigh		
lock		
logoff		
md		
mem		
mkdir		
mklink		
mode		
more		
move		

msav		
msd		
mscdex		
nbtstat		
net		
netsh		
netstat		
nlsfunc		
nslookup		
path		
pathping		
pause		
ping		
popd		
power		
print		
prompt		
pushd		
qbasic		
rd		
ren		
rename		
rmdir		
robocopy		
route		
runas		
sc		
scandisk		
scanreg		
set		
setlocal		
setver		

share		
shift		
shutdown		
smartdrv		
sort		
start		
subst		
switches		
sys		
telnet		
time		
title		
tracert		
tree		
Type		
undelete		
unformat		
unlock		
ver		
verify		
vol		
xcopy		

Practical No 3

Objective: To learn Microsoft office Word

Microsoft office package: …….

……

……

……

……

……

Word Processing: ……..

……

……

……

……

……

Word-wrap: ……..……...

……

Justification: ……...

Adjustment: ……

Alignment: ……...

Decimal Alignment: ……..

Indents: ……...

Insertion: ……..…...

Overstriking: ……..……

Deletion: …….…

Search and Replace: …….…..

Copying or Cutting: ……..

Boilerplate: ……..

……

Pagination: ……...

Page Numbering: …….

Headers and Footers: …….

……

Footnoting: …….……..

……

Table of Contents and Index Generators:. ……..

Form Letter Merging: ……

……..……

Automatic Spelling Checker and Corrector. ……

……..……

……..……

Exercise on Computer:

1. **How to open the MS word File**
2. **How to save the MS word File**
3. **How to change design and fonts size**
4. **How to Secure Word Document**
5. **Create the Table in the word.**
6. **How to insert the pics into word document**

Practical No 4

Objective: To learn working with MS Excel

MS Excel:

……

……

……

……

……

……

……

Features:

Hyperlink. ……..

……

Clip art: ……..

……

Charts: ……..

……

……

Tables: ……

……

……

Macros: ……..

……

Database: ……..

……

Sorting and filtering. ……..

……

Data validations. ……

Grouping. ……

Page layout. ……

……

Exercise on Computer:

1. **How to insert image, in excel sheet**

2. **How to make a graph using Excel**
3. **How to create the charts in the excel sheet.**

Practical No 5

Objective: To learn working with MS Excel and Formula Bar

MS excel: ……..

……

……

……

……

Exercise on Computer

1. Insert the formula in the excel sheet.

x	2	3	4	7	8
y	8	5	6	71	12

Find the value of z; Z= (x+y), Z= (x-y), Z= (x*y)

Practical No 6

Objective: To perform statistical analysis through MS Excel

Statistical analysis: ……

……

……

……

Exercise on Computer

Find the correlation of given data

x	2	3	4	7	8
y	8	5	6	71	12

Practical No 7

Objective: To learn working with MS PowerPoint

Microsoft power point presentation: ……

……

……

……

……

Design: ……

……

……

……

Animation: ……

……

……

……

……

……

Presentation: ……

……

……

……

Integration: ……

……

……

……

Different menu and Submenu of Power point: ……

……

……

……

……

……

……

……

……

……

Practical No 8

Objective: To make presentation with MS PowerPoint

Exercise on Computer:

1. **Create First Slide using MS Power point**
2. **How to insert animation on slide**
3. **Create the PowerPoint presentation on introduction to computer.**
4. **What is the difference between ppt and pptx.**
5. **How to insert effect on ppt.**

Practical No 9

Objective: To study about MS Access

Microsoft Access …….
……
……
……
……
……
……

Exercise on Computer:

How to create Table in MS Access

Practical No. 10

Objective: To study preparation of Query Wizard

Steps to remember:

Click on the CREATE option in the menu bar

Select on QUERY WIZARD

Select the type of query - SIMPLE QUERY

Select fields and click on NEXT

Enter FINISH and query is created

Exercise on Computer:

Write the Query and Run

Create the Table of your classmates and find the name of the students belongs ……. city

Practical No. 11

Objective: To study about computer programming fundamentals, steps to design and develop a computer program, introduction to algorithm and flowchart and their representation.

Algorithm: ……

……

……

……

Example: Algorithm to find sum of two numbers:

Step1: BEGIN

Step2: READ a, b

Step3: ADD a and b and store in variable c

Step4: DISPLAY c

Step5: STOP

Flowchart: ……

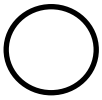
……

……

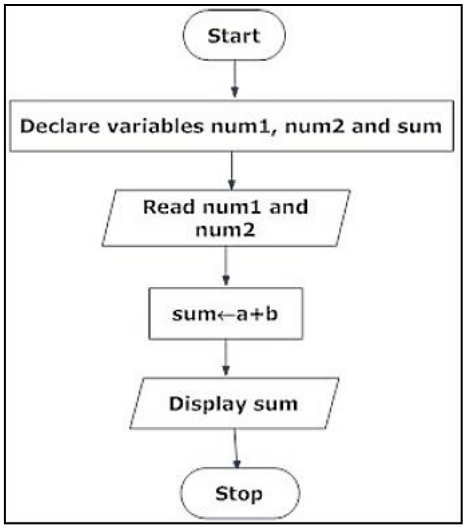
……

Symbols Used in Flowchart:

Symbol	Purpose	Description
→	……	……
▭	…… …… ……	…… …… ……
▭	…… …… ……	…… …… ……
▱	…… …… ……	…… …… ……
◇	…… …… ……	…… …… ……

	…… …… ……	…… …… ……
	…… …… ……	…… …… ……
	…… …… ……	…… …… ……

Example: Flowchart to find sum of two numbers:



Problem 1: Write an algorithm to subtract two numbers.

……

……

……

……

……

……

……

……

……

……..

Problem 2: Draw a flowchart to subtract two numbers.

Practical No. 12

Objective: To introduce students to the programming fundamentals of C language-- basic C program structure, declaration and definition of identifiers and variables, operators with examples.

C is a programming language was developed by Dennis M. Ritchie at AT&T"s BELL Laboratory of USA in 1972. Because of its reliability, C is very popular. C is highly portable & it is well suited for structured programming. C program basically consists of the following parts:

- Preprocessor Commands
- Functions
- Variables
- Statements & Expressions
- Comments

Software requirement to run a C program: Windows/Linux operating system installed with Turbo C/C++, Visual Studio.

Let us look at a simple code that will print "Hello World" in C programming language:

Sample Code	Explanation
#include <stdio.h>	<stdio.h> is a preprocessor command used to include stdio.h file before going to actual compilation.
main() {	main () is the main function where the program execution begins.
/* my first program in C */	The next line /*...*/ is ignored by the compiler and it has been put to add additional comments in the program. So such lines are called comments in the program
printf("Hello, World! \n");	printf(...) is a function which displayed the message "Hello, World!" on the screen
return 0; }	return 0; terminates the main() function. and returns the value 0

<u>Some important points to be remembered in C language:</u>

- In a C program, the semicolon **";"** is a statement terminator.
- Comments are like helping text start with /* and terminate with the characters */.
- **Identifier:** A name used to identify a function, variable, or any other user-defined items. An identifier starts with a letter A to Z, a to z, or an underscore '_' followed by zero or more letters, underscores, and digits (0 to 9). Example: myname50,_temp, a_123.
- **Keywords:** There are 32 keywords/reserved words in C. These reserved words should not be used as constants or variables or any other identifier names.
- **Data type in C:**

Type	Size (bits)	Size (bytes)	Range
char	8	1	-128 to 127
unsigned char	8	1	0 to 255
int	16	2	-2^{15} to $2^{15}-1$
unsigned int	16	2	0 to $2^{16}-1$
short int	8	1	-128 to 127
unsigned short int	8	1	0 to 255
long int	32	4	-2^{31} to $2^{31}-1$
unsigned long int	32	4	0 to $2^{32}-1$
float	32	4	3.4E-38 to 3.4E+38
double	64	8	1.7E-308 to 1.7E+308
long double	80	10	3.4E-4932 to 1.1E+4932

Variable: ……

……...

……...

……...

Variable can be declared as follows:

Variable declaration	Data type declared	Name of the variables
int i, j, k;	Integer	i, j, k
char c, ch;	Character	c, ch
float f, salary;	Float	f, salary
double d;	Double	d

Operators: Operator tells the compiler to perform specific mathematical or logical functions. C language consists of following type of operators:

- Arithmetic Operators: add (+), subtract (-), multiply (*), division (/), modulo (%)
- Relational Operators: less than (<), greater than (>), less than equal to (<=), greater than (>=), equal to (==), not equal to (!=),
- Logical Operators: AND (&&), OR (||), NOT (!)

Example 1: C program to Add Two integer numbers:

Program	Output

#include <stdio.h>
int main() {
int number1, number2, sum;
printf("Enter two integers: ");
scanf("%d %d", &number1, &number2);
// calculating sum
sum = number1 + number2;
printf("%d + %d = %d", number1, number2, sum);
return 0;
}
	Enter two integers: 12
	11
	12 + 11 = 23

Example 2: C program to compute Quotient and Remainder.

Program	Output

#include <stdio.h>
int main() {
int dividend, divisor, quotient, remainder;
printf("Enter dividend: ");
scanf("%d", &dividend);
printf("Enter divisor: ");
scanf("%d", &divisor);
quotient = dividend / divisor; // Computes quotient
remainder = dividend % divisor; // Computes remainder
printf("Quotient = %d\n", quotient);
printf("Remainder = %d", remainder);
return 0;
}
	Enter dividend: 25
	Enter divisor: 4
	Quotient = 6
	Remainder = 1

Problem 1: Write a C program to subtract and multiply two float numbers.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Problem 2: Write a C program to divide two float numbers.

.....

.....

.....

.....

.....

.....

.....

.....

.....

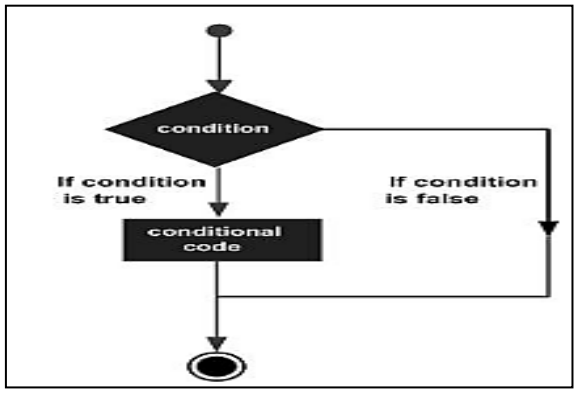
.....

.....

Practical No. 13

Objective: To introduce students to the programming fundamentals of C language-- basic decision structure and loop structure with examples.

Decision making structure in C language: For implementing the decision-making structure programmer specifies one or more conditions to evaluate or test the program. The statements will be executed if the condition determined to be true, and optionally, other statements to be executed if the condition determined to be false. Typical decision-making structure found in most of the programming languages is represented using the following flowchart−



Syntax of "**if...else"** statement in C programming language is as follows:

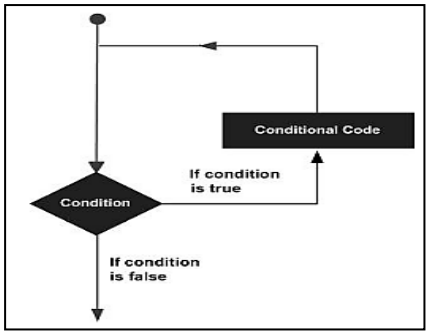
```
if (boolean_expression) {
   /* statement(s) will execute if the boolean expression is true */
} else {
   /* statement(s) will execute if the boolean expression is false */
}
```

Example 1: C program to demonstrate "if…else" statement.

Program	Output

#include <stdio.h>
int main () {
/* local variable definition */
int a = 100;
/* check the boolean condition */
if(a < 20) {
/* if condition is true then print the following */
printf("a is less than 20\n");
} else {
/* if condition is false then print the following */
printf("a is not less than 20\n");
}
printf("value of a is : %d\n", a);
return 0;
}
	a is not less than 20;
	value of a is : 100

Loop structure: Sometimes situation may encounter, when a block of code needs to be executed several numbers of times. In general, statements are executed sequentially: The first statement in a function is executed first, followed by the second, and so on. A loop statement allows to execute a statement or group of statements multiple times. Given below is the general form of a loop statement in most of the programming languages −



A "**for**" loop is a repetition control structure that allows to efficiently write a loop that needs to execute a specific number of times. It is commonly used loop structure in C language. Syntax of "**for**" loop statement in C programming language is as follows:

Syntax	Explanation

for (init; condition; increment) {
statement(s);
}
	The init step is executed first, and only once. This step allows declaring and initializing any loop control variables.
	
	Next, the condition is evaluated. If it is true, the body of the loop is executed. If it is false, the body of the loop does not execute and the flow of control jumps to the next statement just after the 'for' loop.
	
	After the body of the 'for' loop executes, the flow of control jumps back up to the increment statement. This statement allows you to update any loop control variables. This statement can be left blank, as long as a semicolon appears after the condition.

Example 2: C program to demonstrate "for" loop.

Program	Output

#include <stdio.h>
int main () {
int a;
/* for loop execution */
for(a = 10; a < 20; a = a + 1){
printf("value of a: %d\n", a);
}
return 0;
}
	value of a: 10
	value of a: 11
	value of a: 12
	value of a: 13
	value of a: 14
	value of a: 15
	value of a: 16
	value of a: 17
	value of a: 18
	value of a: 19

Problem 1: Draw flowchart and C program to print numbers from 1 to 10.

Problem 2: Write flowchart and C program to Program to calculate the sum of first n natural numbers.

Practical No. 14

Objective: To introduce students to the programming fundamentals of C++ language- basic C ++ program structure, declaration and definition of identifiers and variables, operators, basic decision structure and loop structure with examples.

Software requirement to run a C++ program: Windows/Linux operating system installed with Turbo C++, Visual Studio.

Let us look at a simple code that will print "Hello World" in C++ programming language:

Sample Code	Explanation
#include <iostream>	**<iostream>** is a header that contains necessary or useful information
using namespace std;	**using namespace std;** tells the compiler to use the std namespace
int main() {	**int main()** is the main function where the program execution begins
cout << "Hello World";	**cout << "Hello World"**is a function which displayed the message "Hello, World!" on the screen
return 0; }	**return 0;** terminates the main() function. and returns the value 0

<u>Some important points to be remembered in C++ language:</u>

- In a C/C++ program, the semicolon **";"** is a statement terminator.
- Comments are like helping text start with /* and terminate with the characters */.
- **Identifier and Variable:** Rules of writing identifiers and variables are same as in C language.
- **Data type in C++:** The data type in C++ is same as in C but vary in sizes.
- **Operators:** Types of operator used in C++ language is same as used in C programming.

Example 1: C++ program to Add Two integer numbers:

Program	Output

#include <iostream>
using namespace std;
int main()
{
int firstNumber, secondNumber, sumOfTwoNumbers;
cout << "Enter two integers: ";
cin >> firstNumber >> secondNumber;
sumOfTwoNumbers = firstNumber + secondNumber;
cout << firstNumber << " + " << secondNumber << " = " << sumOfTwoNumbers;
return 0;
}
	Enter two integers: 4
	5
	4 + 5 = 9

Example 2: C++ program to compute Quotient and Remainder.

Program	Output

#include <iostream>
using namespace std;
int main()
{
int divisor, dividend, quotient, remainder;
cout << "Enter dividend: ";
cin >> dividend;
cout << "Enter divisor: ";
cin >> divisor;
quotient = dividend / divisor;
remainder = dividend % divisor;
cout << "Quotient = " << quotient << endl;
cout << "Remainder = " << remainder;
return 0;
}
	Enter dividend: 13
	Enter divisor: 4
	Quotient = 3
	Remainder = 1

Example 3: C++ program to check whether number is even or odd using **"if…else"** statement.

Program	Output

#include <iostream>
using namespace std;
int main()
{
int n;
cout << "Enter an integer: ";
cin >> n;
if (n % 2 == 0)
cout << n << " is even.";
else
cout << n << " is odd.";
return 0;
}
	Enter an integer: 23
	23 is odd.

Example 4: C program to demonstrate "for" loop.

Program	Output

#include <iostream>
using namespace std;
int main () {
// for loop execution
for(int a = 10; a < 20; a = a + 1) {
cout << "value of a: " << a << endl;
}
return 0;
}
	value of a: 10
	value of a: 11
	value of a: 12
	value of a: 13
	value of a: 14
	value of a: 15
	value of a: 16
	value of a: 17
	value of a: 18
	value of a: 19

Problem 1: Write a C++ program to subtract and multiply two float numbers.

……

……

……

……

……

……

……

……

……

……

……

……

……

……

Problem 2: Draw flowchart and C++ program to Program to calculate the sum of first n natural numbers.

Practical No. 15

Objective: To introduce students to the programming fundamentals of C++ language-- object, class, inheritance, polymorphism, abstraction, encapsulation.

The main purpose of C++ programming is to add object orientation to the C programming language. Classes are the central feature of C++ that supports object-oriented programming. A class is used to specify the form of an object and it combines data representation and methods for manipulating that data. The data and functions within a class are called members of the class.

Class definitions in C++

Class definition starts with the keyword **"class"** followed by the class name; and the class body, enclosed by a pair of curly braces. A class definition must be followed either by a semicolon or a list of declarations. For example, the Box class is defined as follows:

```
class Box {
  public:
    double length;   // Length of a box
    double breadth;  // Breadth of a box
    double height;   // Height of a box
};
```

Inheritance in C++

Inheritance is one of the most important concepts in object-oriented programming. Inheritance allows in defining a class in terms of another class. This provides an opportunity to reuse the code functionality and fast implementation time. When creating a class, instead of writing completely the new data members and member functions, the programmer can designate that the new class should inherit the members of the existing class. This existing class is called the **base class**, and the new class is referred to as the **derived class.** The idea of inheritance is to implement **"is a"** relationship. For example, mammal **"is an"** animal, dog **"is a"** mammal hence dog **"is an"** animal as well and so on.

Example 1: C++ program to demonstrate inheritance:

Program	Output

// Base class
class Shape {
  public:
  void setWidth(int w) {
  width = w;
  }
  void setHeight(int h) {
  height = h;
  }
  protected:
  int width;
  int height;
};
// Derived class
class Rectangle: public Shape {
	Total area: 35

```
    public:
      int getArea() {
        return (width * height);
      }
  };
  int main(void) {
    Rectangle Rect;
     Rect.setWidth(5);
    Rect.setHeight(7);
    // Print the area of the object.
    cout << "Total area: " << Rect.getArea() << endl;
    return 0;
  }
```

Problem 1: In forestry, there is *Pinaceae* family having some specific characteristics. *P. roxburghii* and *Pinus wallichiana* are two forest trees under the pine family. Write a C++ program of inheritance for the same with flowchart.

Practical No. 16

Objective: To introduce students to the programming fundamentals of Java language-- basic Java program structure, declaration and definition of identifiers and variables, operators, basic decision structure and loop structure with examples.

Java is a high level, robust, object-oriented and secure programming language. Java was developed by Sun Microsystems (which is now the subsidiary of Oracle) in the year 1995. James Gosling is known as the father of Java. Java uses compiler and interpreter both. Java source code is converted into byte code at compilation time. The interpreter executes this byte code at runtime and produces output. Java is interpreted that is why it is platform independent.

Software requirement to run a C program: Windows/Linux operating system installed with Java Development Kit (JDK), Java Runtime Environment (JRE).

Let us look at a simple code that will print "Hello World" in Java programming language:

Sample Code	Explanation

public class Test{
public static void main(String args[]){
System.out.println("Hello World");
}
}
	• **return 0;** terminates the main() function. and returns the value 0.
	• **public**' is access modifier.
	• **Test** is class name.
	• **public static void main(..) :** main method with void return type.
	• **System.out.println(..):** Print statement.

<u>Some important points to be remembered in Java language:</u>

- Every statement in java is terminated by '**;**'.
- Reserved keywords cannot be used as identifier.
- Rules of writing identifiers and variables are same as in C language.
- File name should be same as class name of the main method.
- There may be many methods in Java programme but there is always a main method.
- Java comments (// single line comment or /* multi line comment */) are not executed by the compiler & interpreter.

Example 1: Java program to add two integer numbers:

Program	Output

public class AddTwoIntegers {
public static void main(String[] args) {
int first = 10;
int second = 20;
int sum = first + second;
System.out.println("The sum is: " + sum);
}
}
	Enter two numbers: 10 20
	The sum is: 30

Example 2: Java program to compute quotient and remainder.

Program	Output

public class QuotientRemainder {
public static void main(String[] args) {
int dividend = 25, divisor = 4;
int quotient = dividend / divisor;
int remainder = dividend % divisor;
System.out.println("Quotient = " + quotient);
System.out.println("Remainder = " + remainder);
}
}
	Quotient = 6
	Remainder = 1

Example 3: Java program to check whether number is even or odd using **"if…else"** statement.

Program	Output

import java.util.Scanner;
public class EvenOdd {
public static void main(String[] args) {
Scanner reader = new Scanner(System.in);
System.out.print("Enter a number: ");
int num = reader.nextInt();
if(num % 2 == 0)
System.out.println(num + " is even");
else
System.out.println(num + " is odd");
}
}
	Enter a number: 12
	12 is even

Example 4: Java program to demonstrate "for" loop.

Program	Output

Live Demo
public class Test {
public static void main(String args[]) {
for(int x = 10; x < 20; x = x + 1) {
System.out.print("value of x : " + x);
System.out.print("\n");
}
}
}
	value of a: 10
	value of a: 11
	value of a: 12
	value of a: 13
	value of a: 14
	value of a: 15
	value of a: 16
	value of a: 17
	value of a: 18
	value of a: 19

Problem 1: Write a Java program to subtract and multiply two float numbers.

....

....

....

....

....

....

....

....

....

Problem 2: Write flowchart and Java program to program to calculate the sum of first n natural numbers.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Practical No. 17

Objective: To introduce students to the programming fundamentals of Java language-- object, class, inheritance, polymorphism, abstraction, encapsulation with examples.

Object and **class** are same as defined in the C++ practical manual. **Inheritance** in Java is a mechanism in which one object acquires all the properties and behaviors of a parent object. Code reusability is achieved through this.

Example 1: Java program to demonstrate inheritance:

Program	Output

class Employee{
float salary=40000;
}
class Programmer extends Employee{
int bonus=10000;
public static void main(String args[]){
Programmer p=new Programmer();
System.out.println("Programmer salary is:"+p.salary);
System.out.println("Bonus of Programmer is:"+p.bonus);
}
}
	Programmer salary is:40000.0
	
	Bonus of Programmer is:10000

Polymorphism in Java is a concept by which we can perform a single action in different ways. There are two types of polymorphism in Java: compile-time polymorphism and runtime polymorphism. We can perform polymorphism in java by method overloading and method overriding.

- **Method Overloading**: If a class has multiple methods having same name but different in parameters, it is known as Method Overloading.
- **Method Overriding**: If subclass (child class) has the same method as declared in the parent class, it is known as method overriding in Java.

Example 2: Java program to demonstrate polymorphism:

Program	Output

// Java program to demonstrate working of method overloading in Java.
public class Sum
{
// Overloaded sum(). This sum takes two int parameters
public int sum (int x, int y)
{
return (x + y);
}
// Overloaded sum(). This sum takes three int parameters
public int sum (int x, int y, int z)
{
return (x + y + z);
}
	30
	
	60
	
	31.0

```
// Overloaded sum(). This sum takes two double parameters
  public double sum (double x, double y)
  {
return (x + y);
}
  public static void main (String args[])
  {
Sum s = new Sum (); // create a object 's' of Sum class
System.out.println (s.sum (10, 20)); // calling of object and method sum
System.out.println (s.sum (10, 20, 30));
System.out.println (s.sum (10.5, 20.5));
 }
   }
```

Abstraction is a process of hiding the implementation details and showing only functionality to the user. **Encapsulation** in Java is a process of wrapping code and data together into a single unit, for example, a capsule which is mixed of several medicines.

Problem 1: In forestry, there is *Pinaceae* family having some specific characteristics. *P. roxburghii* and *Pinus wallichiana* are two forest trees under the pine family. Write a Java program of inheritance for the same with flowchart.