**A1.**

**a)- in notebook**

**b)- in notebook**

**A2.**

a)-gives the remainder

b)-return the integer value of a given string input

c)- used to define a constant

d)- to have two or more methods having same name but different argument lists within the same class.

e)- is an increment operator that increases the value of operand by 1

**A3.**

a)- illegal

b)- legal

c)- illegal

d)- illegal

e)- legal

f)- illegal

**A4.**

a)-

if(mark<30)

{

System.out.println(“mark is below 30”);

}

b)-

public static char checkInfo(int w, int y)

c)-

point=point-1;

bonus=bonus+point;

**A5.-**

**a)- Notebook**

**b)-**

final int size=200;

c)-

int[] amount=new int[size];

d)-

for(int i=0;i<amount.length;i++)

{

System.out.println(amount[i]);

}

**A6.**

byte, short, int, long

**B1.**

**a)**

import java.io.\*;

class Gender

{

public static void main(String[] args)throws Exception

{

char gender;

System.out.println("Enter the gender: ");

gender=(char)System.in.read();

//absorb the EnterKey

System.in.read();

System.in.read();

switch(gender)

{

case 'm':

case 'M':

System.out.println("The sex is Male");

break;

case 'f':

case 'F':

System.out.println("The sex is Female");

break;

default:

System.out.println("Invalid entry.");

}

}

}

**b)**

- flowchart representing (a) logic

**B2.­**

**--Chapter 8**

**a)**

**-**

**b)**

Two features of External sort-

1. Data to be sorted cannot be fitted in the main memory all at once.
2. Data must reside in a secondary storage.

**c)-**

public static void BubbleSort(int[] data)

{

//temporary storage

int temp;

for(int pass=1;pass<N.length;pass++)

{

for(int i=0;i<N.length-1;i++)

{

if(N[i]>N[i+1])

{

temp=N[i];

N[i]=N[i+1];

N[i+1]=temp;

}

}

}

}

**d)-**

advantage: one of simplest method to use and easy to understand.

disadvantage: iteration process still continues even when the items are sorted in the required order.

**e)-** compareTo()

**B3.**

**a)**

**b)**

**i)-** init(), start(), stop(), destroy()

**ii)-** used to draw horizontal, vertical and diagonal lines.

iii)-

drawLine(int x1, int y1, int x2,int y2);

* + x1 - the first point's *x* coordinate.
  + y1 - the first point's *y* coordinate.
  + x2 - the second point's *x* coordinate.
  + y2 - the second point's *y* coordinate.

Eg.

g.drawline(10,10,50,50);

**c)**

**-**

import java.applet.\*;

import java.awt.\*;

public class DrawString extends Applet

{

public void paint(Graphics g)

{

g.setColor(Color.blue);

g.drawString(“hello”,55,35);

}

}