**A1.**

**a)-** TRUE

**b)-** TRUE

**c)-**FALSE

**d)-** FALSE

**e)-** TRUE

**f)-** FALSE

**g)-** TRUE

**h)-** FALSE

**i)-** TRUE

**j)-**FALSE

**A2.**

**a)-** 6

**b)-** 12

**c)-** 0

**d)-** false

**e)-** 13

**A3.**

1. length**()**
2. start**()**
3. %
4. new
5. source code
6. final **---** constant
7. selection
8. boolean
9. setText**()**
10. counter-controlled or definite

**A4.**

* The purpose of using the stop() method is to pause any applet program such as animation, audio files etc. when the page is not currently view i.e. when user leaves the page.

**­**

**A5.**

**a)**

**-** A container object is an object of java.awt.Container class that provides an area

where other GUI components and other containers can be placed inside it.

**b)**

**-** Any 3 of following:

Window, Frame, Panel, ScrollPane, Dialog, FileDialog

**A6.**

**a)**

**-** javac Examination.java

**b)**

**-** java Examination

**A7.**

**-**

* Any 3 of the following:
  1. Arithmetic operator. Eg. +
  2. Relational or comparison operator. Eg. <
  3. Logical operator. Eg. &&
  4. Assignment operator. Eg. =
  5. Increment/Decrement operator. Eg. ++

**B1.**

**a)**

**i)-**

float[] product=new float[100];

**ii)-**

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

{

if(product[i]>12)

{

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

}

}

**b)**

**-**

int i=11;

do

{

if((i%2)==1)

{

System.out.println(i);

}

i++;

}while(i<=77);

**c)**

**-**

import java.io.\*;

class ReadInteger

{

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

{

BufferedReader input=new BufferedReader(new InputStreamReader(System.in));

System.out.println("Enter a number:");

int num=Integer.parseInt(input.readLine());

if(num>50)

{

System.out.println("The number: " + num + " is above 50");

}

else

{

if(num<50)

{

System.out.println("The number: " + num +" is below 50");

}

}

}

}

**B2.­**

**a)**

**i)-** This method compares this String to another String, ignoring case considerations.

**ii)-**

Returns an integer value bycomparing two strings lexicographically.

The comparison is based on the Unicode value of each character in the strings.

**OR**

The method compareTo() is used for comparing two strings lexicographically. Each character of both the strings is converted into a Unicode value for comparison. If both the strings are equal then this method returns 0 else it returns positive or negative value. The result is positive if the first string is lexicographically greater than the second string else the result would be negative.

**iii)**- This method is used to add component in the container in java applet.

**b)**

**-**

**i)**

* **public:** this method can be accessed by any class in any package.
* **static:**this method can be called without creating an instance of this class.
* **float:** this method return a value of float data type.
* **median:** it is the name of this method**.**
* **parameter:** this method takes two parameters a and b of float data type.

**OR**

The method name is median which can be accessed by any class in any package

without creating an instance of this class. This method takes two parameters a and b of

float data type and return a float value.

**ii)**

**-**

* **public:** this method can be accessed by any class in any package.
* **static:**this method can be called without creating an instance of this class.
* **boolean:** this method return a value of boolean data type i.e. either true or false
* **check:** it is the name of this method**.**
* **parameter:** this method takes one parameter w of integer data type.

**iii)**

**-**

* **public:** this method can be accessed by any class in any package.
* **static:**this method can be called without creating an instance of this class.
* **void:** this method does not return anything.
* **showing:** it is the name of this method**.**
* **parameter:** this method takes no parameters.

**c)**

**-**

**i)**

**-**

Area ourArea=new Area();

**ii)**

**-**

public Area(int h, int w)

{

height=h;

width=w;

}

**d)**

-

**i)**

-

**Error:**

1. The return type is **float**.
2. The variable a and b is not defined inside the method body.

**Correction:**

1. Replace float with **void** return type.
2. Replace a and b with x and y.

OR

**Error:**

1. the return type is **float** and no return statement is used inside the body.
2. The variable a and b is not defined

**Correction:** replace *float* with **int** return type and insert the statement ‘*return result*’ before the **closing bracket** of **method body**.

**ii)**

-

**Error:** in calculation (2nd statement) inside the method body, a backslash

is used.

**Correction:** remove **backslash** and insert **forward slash**.

**e)**

-

Encapsulation, Inheritance and Polymorphism.

**B3.**

**a)**

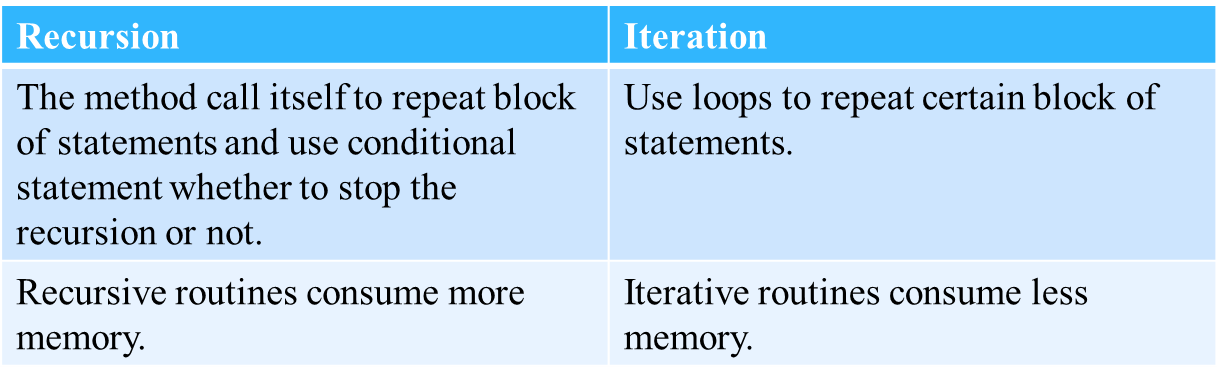
**i)**

**-**

It is the process where a method continually call itself until a defined point of termination.

**ii)**

**-**

****

**iii)**

* Page 8-12

**b)**

**i)**

**-**

* + **drawRect() :** It is used to draw a rectangle or a square.
  + **drawOval():**It is used to draw a circle or ellipse.
  + **drawArc():**It is used to draw an arc.

**ii)**

-

import java.applet.\*;

import java.awt.\*;

public class DrawCircle extends Applet

{

public void paint(Graphics g)

{

g.drawOval(50,65,200,200);

}

}