**A1.**

**a)-** INVALID

**b)-**VALID

**c)-** INVALID

d)- VALID

**e)-** INVALID

**f)-** INVALID

g)- VALID

h)- VALID

i)- INVALID

j)- INVALID

**A2.**

**a) –** Primitive types and Reference types

**b)-**

**i)-** Illegal

ii)- Legal

iii)- Legal

iv)- Legal

v)- Legal

vi)- Illegal

**A3.**

a)- 3

b)- 20

c)- 24

d)- -3

e)- 15

f)- true

g)- true

**A4.**

a)-

if(mark!=30)

{

System.out.println(mark + " is not equal to 30.");

}

b)- public static double getArea(float a,float b)

c)- balance=balance+20;

**A5.-**

switch(status)

{

case 1:

point=1;

break;

case 2:

case 3:

point=2;

break;

default:

point=3;

}

**B1.**

**a)**

i)- Selection

ii)- Iteration

iii)- Selection

**b)**

**-** Page 5-13

**c)**

**- i)**

**Error:** A equal to sign (=) after less than is missing in the condition.

**Correction:** Replace less than operator (<) with less than or equal to operator (<=) in the while condition.

**ii)**

**- Error:** In the counter progression, the loop control variable is increased i.e. x+=2

**Correction:** Decrement the loop control variable in the counter progression i.e. x-=2

**d)**

i)

- public static int sum()

{

return 0;

}

ii)-

public static int sum(int x,int y)

{

int total;

total=x+y;

return total;

}

iii)-

public static void main(String[] args)

{

int a=sum();

int b=sum(10,20);

}

iv)-

Method Overloading:

Method Overloading is a feature that allows a class to have two or more methods having same name, if their argument lists are different.

**e)-**

**Try it yourself**

**B2.­**

**a)-**

**b)-**

**c)-** final int MAX\_HOUR=12;

d)-

i)-

minimum value: -32768

maximum value: 32767

ii)-Reference data type

e)-Notebook || Page 2-13

f)-

**B3.**

**a)**

i) Page 9-3

ii) Button, Label, TextField

**b)**

- FlowLayout,GridLayout, BorderLayout (Page9-20)

**c)**

**i)-** Ellipse or Circle

ii)- drawOval(int x,int y, int width,int height)

x& y:…………………

width & height:……………

iii)-

import java.applet.\*;

import java.awt.\*;

public class DrawOval extends Applet

{

public void paint(Graphics g)

{

setBackground(Color.red);

g.setColor(Color.yellow);

g.drawString("this is an oval",50,10);

g.drawOval(50,50,200,180);

}

}