COMPUTER APPLICATIONS
(Theory)

Question 1

a) In computer programming, the **operator precedence** is a rule used to clarify which procedures should be performed first in a given mathematical expression. When two operators share an operand the operator with the higher precedence goes first. For example, 1 + 2 * 3 is treated as 1 + (2 * 3), whereas 1 * 2 + 3 is treated as (1 * 2) + 3 since multiplication has a higher precedence than addition. When two operators with the same precedence the expression is evaluated according to its associativity. For example $x = y = z = 17$ is treated as $x = (y = x = 17)$, leaving all three variables with the value 17, since the = operator has right-to-left associativity (and an assignment statement evaluates to the value on the right hand side). On the other hand, 72 / 2 / 3 is treated as (72 / 2) /3 since the / operator has left-to-right associativity.

b) A **literal** is a notation for representing a fixed value in source code. Almost all programming languages have notations for atomic values such as integers, floating-point numbers, strings, and booleans; some also have notations for elements of enumerated types and compound values such as arrays, records and objects. Literals are often used to initialize variables, ex : **int** a=1;

String s="cat";

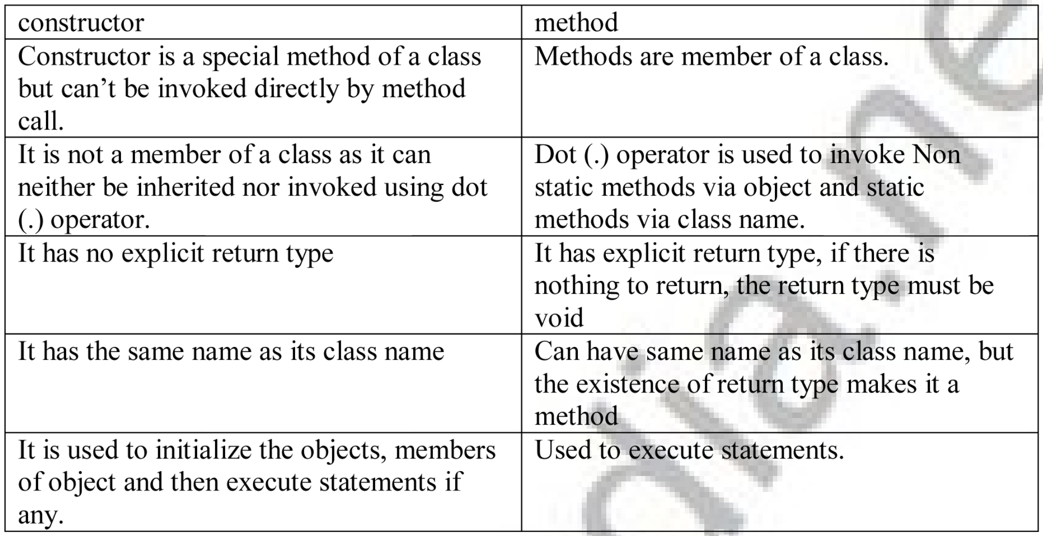
c) (i) A superclass and a subclass :

These terms points to the inheritance concept of java. In the Java language, classes can be *derived* from other classes, thereby *inheriting* fields and methods from those classes. A class that is derived from another class is called a *subclass* (also a *derived class*, *extended class, or child class*). The class from which the subclass is derived is called a *superclass*

(ii) *The act of representing essential features without including background details*

abstraction is the process by which data and programs are defined with a representation similar in form to its meaning (semantics), while hiding away the implementation details. Abstraction captures only those details about an object that are relevant to the current perspective.

d)



constructor	method
Constructor is a special method of a class but can't be invoked directly by method call.	Methods are member of a class.
It is not a member of a class as it can neither be inherited nor invoked using dot (.) operator.	Dot (.) operator is used to invoke Non static methods via object and static methods via class name.
It has no explicit return type	It has explicit return type, if there is nothing to return, the return type must be void
It has the same name as its class name	Can have same name as its class name, but the existence of return type makes it a method
It is used to initialize the objects, members of object and then execute statements if any.	Used to execute statements.

e) 1) Double *x*=15.2;

Int y=(int) *x*;

conversions, specially those that imply a different interpretation of the value, require an explicit conversion.

2) Int *x*=12;

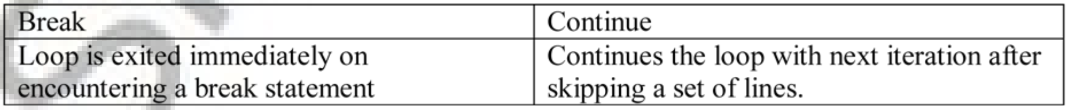
Long y=*x*;

Implicit conversions do not require any operator. They are automatically performed when a value is copied to a compatible type.

Question 2

a) Boolean, Character

b)



Break	Continue
Loop is exited immediately on encountering a break statement	Continues the loop with next iteration after skipping a set of lines.

c) The length of a character array can be found using length method.

Ex : char arr[]=new char [10];

Intlengtharr=arr.length;

The lengtharr will give the size of the arr array.

To find the length of a string object length () method is to be used

Ex: String s="kerala";

Intlength=s.length();

d) i) void

ii) this keyword

e)An *exception* is an event, which occurs during the execution of a program, that disrupts the normal flow of the program's instructions.

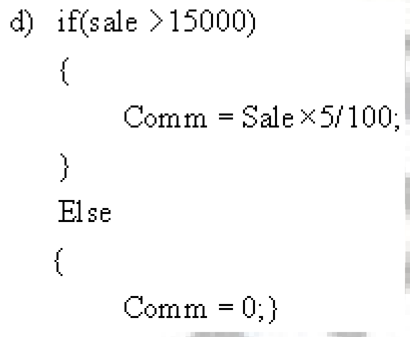
Question 3

a) digital mp4=new digital();

b) Str1="d manners";

Str2="goodd manners";

c) Encapsulation is the technique of making the fields in a class private and providing access to the fields via public methods. If a field is declared private, it cannot be accessed by anyone outside the class, thereby hiding the fields within the class.



```
d)  if(sale >15000)
    {
          Comm = Sale×5/100;
    }
    Else
    {
          Comm = 0;}
```

e) the loop will be executed times and the value returned will be 15.

f) i) The method determines whether the specified char value is a white space which includes space, tab or new line, the data type will be Boolean.

ii) double will be the dat type returned
g) (u * t)+(f*(Math.pow(t,2))/)/2;
h) *x*=343.0
y=5.0
i)18
j) (i) Scanner in = new Scanner (System.in);
Inta = in.nextInt()
(ii) Scanner in = new Scanner(System.in);
String s = in.nextLine();

Section B

Question 4

```
Class FruitJuice
{
 Intproduct_code,pack_size,product_price;
 String flavor,pack_type;
 FruitJuice()
 {
 product_code=0;
 pack_size=0;
 product-price=0;
 flavor="";pack_type="";

}

Public void input()
{

 Scanner in = new Scanner(System.in);

 System.out.println("enter product code");

 product_code = in.nextInt();
 System.out.println("enter pack size");

 pack_size = in.nextInt()
 System.out.println("enter product price");

 product_price = in.nextInt()
 System.out.printIn("enter flavor");
```

```
                flavor = in.nextLine();
        System.out.println("enter pack_type"):

                pack_type = in.nextLine();
        }
        Public void discount()
        {
                Product_price=product_price-10;
        }
        Public void display ()
        {
                System.out.println("product code: "+product_code);
                System.out.println("flavour:"+flavour);
System.out.println("pack type: "+pack_type);
System.out.println("pack size: "+pack_size);
System.out.println("product price: "+product_price);
        }
publicstaticvoid main (String args[]){

FruitJuiceob=new FruitJuice();
Ob.input();
Ob.discount();
Ob.display();

}
}
```

Question 5

```
publicclasssampl {
        staticintisbn[]=newint[10];staticintsum=0,temp=0;
publicstaticvoid main(String args[]){
        read();
}
publicstaticvoid read()
 {Scanner in = newScanner(System.in);

System,out.println("enter isdn number");
for(int i=0;i<isbn.length;i++)
        isbn[i] = in.nextInt();
        if(isbn.length==10){
                for(int i=0,i<isbn.length;i++){
                        sum +=(i+1)*(isbn[i]);

                }
                if(( sum % 11)==0){
                        System.out.println("ISBN is legal");
                }
                else
                        System.out.println("illegal isbn");
        }
        else
                System.out.println("illegal ISBN");


 }
}
```

Question 6

```
publicclasssampl {
        static String word,newword;staticintcount=0;
publicstaticvoid main(String args[]){
        Scanner in = newScanner(System.in);
        System.out.println("enter the word");

                word = in.nextLine();
                word=word.toUpperCase();
                System.out.println(word);
                for(int i=0;i<word.length();i++){
                        if(count==0){

        if((word.charAt(i)=='A'||(word.charAt(i)=='E'||(word.charAt(i)=='I'||
(word.charAt(i)=='O')||(word.charAt(i)=='U')){
                                        newword=word.substring(i, word.length());
                                        count++;
                                                                }}
                                if(count>0){
                                newword+=word.substring(0,i);
                                newword+="AY";

                                System.out.println(newword);
                                break;
```

```
                                        }
                                }
}
}
```

Question 7

```
publicclasssampl {
        staticintarr[]=newint[10],temp=0;
        publicstaticvoid main(String args[]){
         Scanner in = newScanner(System.in);
        System.out.println("enter the array");

                 for(int i=0;i<arr.length;i++)
                {
                         arr[i]-in.nextInt();
                }
                for(int i=0;i<arr.length;i++)
                {for(int j=i+l;j<arr.length;j++){
                        if(arr[i]<arr[j])
                        {
                                temp=arr[i];
                                arr[i]=arr[j];
                                arr[j]= temp;
                         }
                }
                }
                System.out.println("sorted array in descending order is ");
                for(int i=0;i<arr.length;i++)
                {
                System.out.println(arr[i]);
                }
}
}
```

Question 8

```
publicclasssampl {
        staticintarr[]=newint[10];
        publicstaticvoid main(String args[]){
                System.out.println("enter range");
                Scanner in = newScanner(System.in);
```

```
            int n = ib.nextInt();
            //calling series method with one parameter
            System.out.println(series(n));
            System.out.println("enter base value");
            int a = in.nextInt();
            //calling series method with two parameters
            System.out.println(series(a,n));
}
      publicstaticdouble series(double n)
      {               int sum=0;
            for(int i=l;i<=n;i++){
            sum+=l/i;
      }
            return sum;
      }
      publicstaticdouble series(doublea,double n)
      {                     int sum=0;
            for(int i=i;i<=n;i+=3){
            sum+=(i/Math.pow(a,i+l));
            System.out.println(sum);
      }
            return sum;
      }
}
```

Question 9

```
publicclasssampl
      staticintarr[]=newint[10];
      publicstaticvoid main(String args[]){
            System.out. println("enter number");
            Scanner in = newScanner(System.in);
            int x = in.nextInt();
            System.out.println("        Menu");
            System.out.println("   1. check composite");
            System.out.println("   2. check smallest digit");
            System.out.println("enter 1 to check composite and 2 to get smallest
            digit of a number");
            int op = in.nextInt();
            switch(op){
```

```
            case 1 :
                        booleanisComposite = false;
                                for (int i =2; i <x; i++) {
                                if (x % i ==0) {
                                isComposite = true;
                                }
                                }
                                 if (isComposite) {
                                System.out.println(x + " is a composite number");
                                }
                                else {
                                System.out.println(x + " is not a composite number ");
                                }
                                break;
            case 2 :
                    int temp=x ;
                    int min=x%10;
                    while(x>0)
                      {
                    int a=x %10;
                    if(a<min)
                    min=a;
                    x=x /10;
                    }
                    System.out.println("smallest digit of the integer "+temp+"is +min);
                    break;
                    default:
                            System.out.println("incorrect choice");
                            break;
             }


             }
}
```