ARGUMENT PASSING

Information can be passed to methods as parameters. Parameters act as variables inside the method.

Parameters are specified after the method name, inside the parentheses. The multiple arguments are separated by comma.

Types of parameters:

- **Formal Parameter :** A variable and its type as they appear in the prototype of the function or method. **Syntax:**
  function_name(datatype variable_name)

- **Actual Parameter :** The variable or expression corresponding to a formal parameter that appears in the function or method call in the calling environment.
  Syntax:
  func_name(variable name(s));

Important method of parameter passing

- **Pass By Value:** Changes made to formal parameters do not get transmitted back to the caller. Any modifications to the formal parameter variable inside the called function or method affect only the separate storage location and *will not be reflected* in the actual parameter in the calling environment. This method is also called as call by value.

Example:

CALL BY VALUE

```
class CallByValue
{
void Example(int x, int y)
 {
    x++;
     y++;
   }
}
public class Main {
   public static void main(String[] args)
 {
    int a = 10;
    int b = 20;
   CallByValue s= new CallByValue();
System.out.println("a: " + a + " b:" + b);
      s.Example(a, b);
  System.out.println("a: " + a + "  b: "+ b);

    }
}
```

Output:

```
 a: 10  b: 20
 a: 10  b: 20
```

In this example, the value of a and b is passed as a parameter in the function Example is copied to x and y respectively. The changes made in copy(member function) are not reflected in the actual parameter i.e. a and b.

- **Call by reference:** Changes made to formal parameter do get transmitted back to the caller through parameter passing. Any changes to the formal parameter are reflected in the actual parameter in the calling environment as the formal parameter receives a reference (or pointer) to the actual data.

EXAMPLE:

CALL BY REFERENCE

```
 class Call_Reference
 {
    int a, b;
Call_Reference(int x, int y)
 {
      a = x;
      b = y;
    }
void Change(Call_Reference obj)
   {
        obj.a += 10;
        obj.b += 20;
 }
}
public class Main {
   public static void main(String[] args)
    {
int a=20;
int b=40;
Call_Reference obj= new Call_Reference(a,b);
System.out.println("a:"+ obj.a+" b:"+obj.b);
obj.Change(obj);
System.out.println("a:"+ obj.a+" b:" +obj.b);
}
}
```

Output:

```
 a: 20  b: 40
 a: 30  b: 60
```

In the above example, while creating object obj for class Call_Reference have passed two arguments ie value of a and b so that constructor gets invoked.Object of class is passed as argument in function Change.Any changes made in formal parameter is reflected in actual parameter as copy of a reference (address) to an object is passed by value to the method, not a copy of the object itself because Java does not copy the object into the memory.

Returning object

Example:1

A method can return any type of data, including objects.For example, in the following program, the createData( ) method returns an object of Data.

```
class Data.
{
   int data1;
   int data2;
}

class CreateData {

   Data createData()
   {
      return new Data();
   }
}

public class Javaapp {

   public static void main(String[] args) {

      CreateData cd = new CreateData();
```

```
        Data d1 = cd.createData();

        d1.data1 = 10;
        d1.data2 = 20;
        System.out.println("d1.data1 : "+d1.data1);
        System.out.println("d1.data2 : "+d1.data2);
    }
}
```

<u>Output</u>:

```
  d1.data1: 10
  d1.data2 :20
```

<u>Example:2</u>

```
class Rectangle {
      int length;
      int breadth;

      Rectangle(int l,int b) {
        length = l;
        breadth = b;
      }

      Rectangle getRectangleObject() {
        Rectangle rect = new Rectangle(10,20);
        return rect;
      }
}

class Main {
      public static void main(String args[]) {
        Rectangle ob1 = new Rectangle(40,50);
        Rectangle ob2;

        ob2 = ob1.getRectangleObject();
        System.out.println("ob1.length : " + ob1.length);
        System.out.println("ob1.breadth: " + ob1.breadth);

        System.out.println("ob2.length : " + ob2.length);
        System.out.println("ob2.breadth: " + ob2.breadth);
        }
}
```

Output

```
ob1.length : 40
ob1.breadth: 50
ob2.length : 10
ob2.breadth: 20
```

PRACTICE PROGRAMS

1.Develop a library interface which has drawbook(), returnbook()(with fine), checkstatus() and reserve book() methods.

```
import java.io.*;
import java.util.*;
import java.text.*;
 interface Library
{
            public void drawbook();
            public void returnbook();
            public void checkstatus();
            public void reservebook();

}
 class Lib implements Library
{
private  int bno;
private String bookname;
private  String authorname;
private String issuedate;
private String returndate;
private String bn;
private String rd;
private String reservedbook;
Scanner sc=new Scanner(System.in);
            public void drawbook()
            {
            System.out.print("enter the book id:");
            bno=sc.nextInt();
            System.out.print("enter the book name:");
            bookname=sc.next();
            System.out.print("enter the author name:");
            authorname=sc.next();
            if(reservedbook==null||(bookname.compareTo(reservedbook)!=0))
            {
            System.out.print("book is not reserved");
            System.out.print("\nbook can be issued");
            System.out.print("\nenter the issued date(yyyy-MM-dd):");
            issuedate=sc.next();
           SimpleDateFormat s=new SimpleDateFormat("yyyy-MM-dd");
            Calendar c=Calendar.getInstance();
            try
            {
                c.setTime(s.parse(issuedate));
            }
            catch(Exception e)
            {

            }
            c.add(Calendar.DAY_OF_MONTH, 7);
            returndate=s.format(c.getTime());
            System.out.print("return the book within one week "+returndate);}
            else
            {
                      System.out.print("book is reserved can not be issued");
            }
            }
            public void returnbook()
            {
             float fine;
             System.out.print("\nenter the book name going to return");
            bn=sc.next();
            System.out.print("enter the date of return(yyyy-MM-dd):");
            rd=sc.next();
            if(rd.compareTo(returndate)<=0)
            {
                      System.out.print("book returned on correct time");
            }
            else
            {
                      SimpleDateFormat s=new SimpleDateFormat("yyyy-MM-dd");
                      try
                      {Date d1=s.parse(returndate);
```

```
                Date d2=s.parse(rd);
                long diff=d2.getTime()-d1.getTime();
                float no_of_dates_diff=(diff/(1000*60*60*24));
                System.out.print("no of dates delay:"+no_of_dates_diff);
                fine=no_of_dates_diff*5;
                System.out.print("\npay fine rs."+fine);
                }
                catch(Exception e)
                {

                }
        }
}
        public void reservebook()
        {
                String reserve;long bID;String author;
                System.out.print("enter the book details to be reserved: ");
                System.out.print("\nenter book id:");
                bID=sc.nextLong();
                System.out.print("enter book name:");
                reserve=sc.next();
                System.out.print("enter author name:");
                author=sc.next();
                reservedbook=reserve;
        }
        public void checkstatus()
        {
                if(bn==null||(bn.compareTo(bookname)==0))
                {

                        System.out.print("\nbook id:"+bno);
                        System.out.print("\nbook name:"+bookname);
                        System.out.print("\nbook author name:"+authorname);
                        System.out.print("\nbook returned on:"+rd);
                }
                else
                {
                        System.out.print("book has not been returned yet");
                }

        }
}
```

```
public class main {

                    public static void main(String args[])
                    {

                              Lib l=new Lib();
                              int choice;
                              Scanner sr=new Scanner(System.in);
                              do
                              {System.out.print("\n1.drawbook\n2.return book\n3.reserve particular
book\n4.checkstatus\n5.exit");
                              System.out.print("\nenter choice:");
                               choice=sr.nextInt();
                              switch(choice)
                              {case 1:l.drawbook();
                              break;
                              case 2:l.returnbook();
                              break;
                              case 3:l.reservebook();
                              break;
                              case 4:l.checkstatus();
                              break;
                              case 5:break;}
                    }while(choice!=5);
          }
}
```

2.

```
import java.util.Scanner;
class Bank
{
          private String name;
          private long acc_no;
          private char type_of_acc;
          private float balc;
          private float bals;
          Scanner sc=new Scanner(System.in);
          public void Newaccount( )
          {
                    System.out.print("Enter the name:");
```

```
		name=sc.next();
		System.out.print("Enter the account number:");
		acc_no=sc.nextLong();
		System.out.print("Type of account(c/s):");
		type_of_acc=sc.next().charAt(0);
		if(type_of_acc=='c')

		{
			System.out.print("minimal balance of c is rs.3000");
			balc=3000;

		}
		else if(type_of_acc=='s')
		{
			System.out.print("minimal balance of s is rs.1000");
			bals=1000;
		}

	}
	public void deposit()
	{
		long accno;
		long amt;
		System.out.print("\nEnter the account number:");
		accno=sc.nextLong();
	if(accno==acc_no)
		{	System.out.print("Enter the amount to be deposit:");
		amt=sc.nextLong();
		if(type_of_acc=='c')
		{balc=balc+amt;
		}
		else if(type_of_acc=='s')
		{
			bals=bals+amt;
		}}
		else
		{
			System.out.print("account not found");
		}

		//System.out.print("current balance "+bal);
```

```
}
public void withdrawal()
{
        long amount;
        long accno;

        System.out.print("Enter account number:");
        accno=sc.nextLong();
        if(accno==acc_no)
        {if(type_of_acc=='c')
        {
                if(balc>=3000)
                {System.out.print("Enter the amount to be withdrawn:");
                amount=sc.nextLong();
                balc=balc-amount;
                System.out.print("Current balance"+balc);
                }
                else
                {
                        System.out.print("Amount can not be withdrawn:");
                }
        }
        else if(type_of_acc=='s')
        {
                if(bals>=1000)
                {
                        System.out.print("Enter the amount to be withdrawn:");
                        amount=sc.nextLong();
                        bals=bals-amount;
                        System.out.print("Current balance "+bals);
                }
                else
                {
                        System.out.print("Amount can not be withdrawn:");
                }
        }}
        else
        {
                System.out.print("account not found");
        }
}
public void display()
```

```
	{
		System.out.print("\nName:"+name+"\nAccount Number:"+acc_no+"\nType of 
account:"+type_of_acc);
		if(type_of_acc=='c')
		{
			System.out.print("\nBalance of current account: "+balc);
		}
		else if(type_of_acc=='s')
		{
			System.out.print("\nBalance of saving account:"+bals);
		}
	}
}

public class main {
	public static void main(String args[])
	{

	Bank b=new Bank();
	Scanner s=new Scanner(System.in);
	int choice;

		do
		{
		System.out.print("\n1.New 
account\n2.Deposit\n3.Withdrawal\n4.Display\n5.Exit");
		System.out.print("\nEnter the choice:");
		choice=s.nextInt();
		switch(choice)
		{
		case 1:b.Newaccount();
		break;
		case 2:b.deposit();
		break;
		case 3:b.withdrawal();
		break;
		case 4:b.display();
		break;
		case 5:break;
		}
		}while(choice!=5);
}
}
```