Authpur National Model Higher Secondary School

Name: - Srinjay Das Gupta

Class- XI- Science

Roll No.-

Subject- Computer Science.

|  |  |  |
| --- | --- | --- |
| Sl.No. | Topic | Page No. |
| 1. | Introduction | 3 |
| 2. | Half-Yearly Assignments | 4- |
| 3. | Annual Assignments |  |
| 4. | Conclusion |  |
| 5. | Bibliography |  |

***INDEX***

*Introduction*

Java is a class-based, object-oriented programming language that is designed to have as few implementation dependencies as possible. It is a general-purpose programming language intended to let application developers write once, run anywhere (WORA), meaning that compiled Java code can run on all platforms that support Java without the need for recompilation.[18] Java applications are typically compiled to *byte code* that can run on any Java virtual machine (JVM) regardless of the underlying computer architecture. The syntax of Java is similar to C and C++, but it has fewer low-level facilities than either of them.

Java was originally developed by James Gosling at Sun Microsystems (which has since been acquired by Oracle) and released in 1995 as a core component of Sun Microsystems' Java platform. The original and reference implementation Java compilers, virtual machines, and class libraries were originally released by Sun under proprietary licenses. As of May 2007, in compliance with the specifications of the Java Community Process, Sun had relicensed most of its Java technologies under the GNU General Public License. Oracle offers its own HotSpot Java Virtual Machine, however the official reference implementation is the OpenJDK JVM which is free open source software and used by most developers including the Eclipse IDE and is the default JVM for almost all Linux distributions.

--x--

**Program: 1.** Write a program to find the digital root of a number.

[Digital root of a number is the single digit that results from the continuous summation of the digits of a

Number and the numbers resulting from each summation .E.g. consider the number 378,

Sum of its digits = 3+7+8 = 18, 1+8 =9

So Digital root of 378 = 9]

**Algorithm:**

1. Start
2. Declare variables n, n1, s and d
3. Read n from user.
4. Initialize variables s=0 , n1= n
5. Repeat the following steps until false
   1. If n1 is not equal to 0, find the remainder of division of n1 by 10
   2. Add the remainders to s.
   3. If s<10, move to Step 6
   4. Else overwrite n1=s and s=0 and continue the steps.
6. Print s as the Digital Root of n.

**Source Code:**

import java.util.\*;

class digital\_Root

{

int digitalRoot(int n)

{

int n1,s=0,d;

n1=n;

while(true)

{

while(n1!=0)

{

d=n1%10;

s=s+d; //Finding the sum of the digits

n1/=10;

}

if(s<10) //Digital root is a one digit no. thus s<10

break;

else

{

n1=s;

s=0;

continue;

}

}

return s;//Returning s as Digital root

}

public static void main(String ars[])

{

int n,dr;

Scanner in=new Scanner(System.in);

System.out.println("Enter a no.");

n=in.nextInt();//Reading n from user

digital\_Root ob=new digital\_Root();//calling digital\_Root()

dr=ob.digitalRoot(n);

System.out.println("The Digital Root of "+n+" is "+dr);

}

}

**Variable Description Table**

|  |  |  |
| --- | --- | --- |
| Variable | Data Type | Function |
| n | int | Accept value from user and a parameter for digital\_Root() |
| n1 | int | Temporary variable to store the value of n |
| s | int | Stores the sum of the digits of n1 |
| d | int | Stores the digits of n1 |
| dr | int | Calls digital\_Root and stores the value returned |

**Program 2:** Write a program to print all the Prime Palindrome numbers in the given range.

**Algorithm:**

1. Start
2. Declare variables start, end, i, j and c.
3. Read start and end from user
4. Run loop from i=start till i=end.
5. Check for prime number
   1. Run a loop from j=1 to j=i
   2. If i%j=1 increment c by 1
   3. Outside the loop check if c=2, then return 1 and overwrite c=0
   4. Else return 0 and overwrite c=0
6. Check for palindrome.
   1. Declare variables n, r and d.
   2. Initialise n=i and r=0
   3. Repeat the following processes until n is 0.
      1. Find the remainder of the division of n by 10 and store in d
      2. Multiply d by 10 and add it to r. store this in r
      3. Store the quotient of the division of n by 10 in n
   4. If r is equal to i return 1
   5. Else return 0
7. If i is Prime as well as palindrome print i.

**Source code:**

import java.util.\*;

class primePalindrome

{

int start,end;

primePalindrome(int a,int b)

{

start=a;end=b;//Initialising data members by parameterised constructor

}

int isPrime(int i)//Checking Prime no.

{

int j,c=0;

for(j=1;j<=i;j++)

{

if(i%j==0)

c++;

}

if(c==2)

return 1;

else

return 0;

}

int isPalin(int i)// Checking for palindrome no.

{

int n,j,r=0,d;

n=i;

while(n!=0)

{

d=n%10;

r=r\*10+d;

n/=10;

}

if(r==i)

return 1;

else

return 0;

}

void generate()//Printing Prime-palindrome no.s

{

int i;

System.out.println("The Prime palindrome no.s b/w "+start+" and "+end+" are ");

for(i=start;i<=end;i++)

{

if(isPrime(i)==1&&isPalin(i)==1)

System.out.println(i+" ");

}

}

public static void main(String ars[])

{

int a,b;

System.out.println("Enter the start and end points");

Scanner in=new Scanner(System.in);

a=in.nextInt();

b=in.nextInt();

primePalindrome ob=new primePalindrome(a,b);

ob.generate();

}

}

**Variable Description Table**

|  |  |  |
| --- | --- | --- |
| Variable | Data Type | Function |
| start | int | To store the starting point |
| end | int | To store the ending point |
| i | int | Loop variable and parameter for isPalin() and isPrime() |
| j | int | Loop variable |
| a | int | Parameter for constructor |
| b | int | Parameter for constructor |
| r | int | Stores the reverse of the no. |
| d | int | Store the digits of |
| c | int | Counts the factors |

**Program 3:** Write a program to take a binary number and convert it into Decimal number.

**Algorithm:**

1. Start
2. Declare variables dec\_out, dec1, dec2, bin\_s, bin\_in, bin\_frac, pi, i, p, ch
3. Initialise dec1=0.0 and dec2=0.0
4. Read bin\_s from user
5. Find the index of ‘.’ In bin\_s and store in ‘pi’
6. Separate the integral and the fractional part of bin\_s into bin\_in and bin\_frac respectively
7. Store the length of bin\_in in p
8. Declare a temporary variable k and initialise it to 0
9. Run a loop from i=(p-1) until i<=1 and repeat the following
   1. Store each character of bin\_in in ch
   2. Check if the integral value of ch is>1
   3. If true terminate the program
   4. Else declare a temporary variable temp to store the integral value of c
   5. Increment the value of dec1 by (temp\*2^k)
   6. Increment k by 1
10. Repeat from Step 7 for bin\_frac but in Step 9 e dec2 will be incremented
11. Concatenate dec1 and dec2 in dec\_out
12. Return dec\_out

**Source Code:**

import java.util.\*;

class binarytodecimal

{

double converter(double bin)

{

double dec\_out,dec1=0.0,dec2=0.0;String bin\_s,bin\_in,bin\_frac;int pi,i,p;char ch;

bin\_s=Double.toString(bin);

pi=bin\_s.indexOf(".");

bin\_in=bin\_s.substring(0,pi);

bin\_frac=bin\_s.substring(pi+1);

p=bin\_in.length();

int k=0;

for(i=(p-1);i>=0;i--)

{

ch=bin\_in.charAt(i);

if(Integer.parseInt(String.valueOf(ch))>1)

{

System.out.println("Invalid Binary No.");

System.exit(0);

}

int temp=Integer.parseInt(String.valueOf(ch));

dec1=dec1+(temp\*Math.pow(2,k));

k++;

}

p=bin\_frac.length();

for(i=0;i<p;i++)

{

ch=bin\_frac.charAt(i);

if(Integer.parseInt(String.valueOf(ch))>1)

{

System.out.println("Invalid Binary No.");

System.exit(0);

}

int temp=Integer.parseInt(String.valueOf(ch));

dec2=dec2+(temp\*Math.pow(2,-(i+1)));

}

dec\_out=dec1+dec2;

return dec\_out;

}

public static void main(String ars[])

{

double bin;

Scanner in=new Scanner(System.in);

System.out.println("Enter a binary no. (fractions included)");

bin=in.nextDouble();

binarytodecimal ob=new binarytodecimal();

System.out.println("The decimal equivalent of "+bin+" is "+(ob.converter(bin)));

}

}

**Variable Description Table**

|  |  |  |
| --- | --- | --- |
| **Variable** | **Data Type** | **Function** |
| dec\_out | double | Store the decimal equivalent |
| dec1 | double | Store the integral decimal |
| dec2 | double | Store the fractional decimal |
| bin\_s | String | Input from user |
| bin\_in | String | Integral part of bin\_s |
| bin\_frac | String | Fractional part of bin\_s |
| pi | int | Index of ‘.’ in bin\_s |
| i | int | Loop variable |
| p | int | Stores the lengths of bin\_in and bin\_frac |
| k | int | Stores the power of 2 |
| temp | int | Store the integral value of ch |
| ch | char | Store each of charac ter of bin\_in and bin\_frac |

**Program 4:** Write a program to accept an Octal number and convert it into Decimal number.

**Algorithm:**

1. Start
2. Declare variables dec\_out, dec1, dec2, oct\_s, oct\_in, oct\_frac, pi, i, p, ch
3. Initialise dec1=0.0 and dec2=0.0
4. Read oct\_s from user
5. Find the index of ‘.’ In oct\_s and store in ‘pi’
6. Separate the integral and the fractional part of oct\_s into oct\_in and oct\_frac respectively
7. Store the length of oct\_in in p
8. Declare a temporary variable k and initialise it to 0
9. Run a loop from i=(p-1) until i<=1 and repeat the following
   1. Store each character of oct\_in in ch
   2. Check if the integral value of ch is>1
   3. If true terminate the program
   4. Else declare a temporary variable temp to store the integral value of c
   5. Increment the value of dec1 by (temp\*8^k)
   6. Increment k by 1
10. Repeat from Step 7 for oct\_frac but in Step 9 (e) dec2 will be incremented
11. Concatenate dec1 and dec2 in dec\_out
12. Return dec\_out

**Source Code:**

import java.util.\*;

class OctToDec

{

double converter(double oct)

{

double dec\_out,dec1=0.0,dec2=0.0;String oct\_s,oct\_in,oct\_frac;int pi,i,p;char ch;

oct\_s=Double.toString(oct);

pi=oct\_s.indexOf(".");

oct\_in=oct\_s.substring(0,pi);

oct\_frac=oct\_s.substring(pi+1);

p=oct\_in.length();

int k=0;

for(i=(p-1);i>=0;i--)

{

ch=oct\_in.charAt(i);

if(Integer.parseInt(String.valueOf(ch))>7)

{

System.out.println("Invalid Octal No.");

System.exit(0);

}

int temp=Integer.parseInt(String.valueOf(ch));

dec1=dec1+(temp\*Math.pow(8,k));

k++;

}

p=oct\_frac.length();

for(i=0;i<p;i++)

{

ch=oct\_frac.charAt(i);

if(Integer.parseInt(String.valueOf(ch))>7)

{

System.out.println("Invalid Octal No.");

System.exit(0);

}

int temp=Integer.parseInt(String.valueOf(ch));

dec2=dec2+(temp\*Math.pow(8,-(i+1)));

}

dec\_out=dec1+dec2;

return dec\_out;

}

public static void main(String ars[])

{

double oct;

Scanner in=new Scanner(System.in);

System.out.println("Enter a octal no. (fractions included)");

oct=in.nextDouble();

OctToDec ob=new OctToDec();

System.out.println("The decimal equivalent of "+oct+" is "+(ob.converter(oct)));

}

}

**Variable Description Table**

|  |  |  |
| --- | --- | --- |
| **Variable** | **Data Type** | **Function** |
| dec\_out | double | Store the decimal equivalent |
| dec1 | double | Store the integral decimal |
| dec2 | double | Store the fractional decimal |
| oct\_s | String | Input from user |
| oct\_in | String | Integral part of oct\_s |
| oct\_frac | String | Fractional part of oct\_s |
| pi | int | Index of ‘.’ in oct\_s |
| i | int | Loop variable |
| p | int | Stores the lengths of oct\_in and oct\_frac |
| k | int | Stores the power of 8 |
| temp | int | Store the integral value of ch |
| ch | char | Store each of charac ter of oct\_in and oct\_frac |

**Program 5:** Write a program to accept a decimal number and convert it into binary number.

**Algorithm:**

1. Declare variables dec, d, bin, dec\_in, dec\_frac and i
2. Initialise bin to null (i.e.,””)
3. Read dec from user
4. Store the integral value of dec in dec\_in
5. Store the fractional value of dec in dec\_frac
6. Execute the following steps until dec\_in=0
   1. Store the remainder of the division of dec\_in by 2 in d
   2. Append d to bin
   3. Divide dec\_in by 2
7. Execute the following steps from i=0 till i=5
   1. Multiply dec\_frac by 2
   2. Declare a variable frac\_bit and store the integral part of dec\_frac(multiplied by 2) in frac\_bit
   3. Check if frac\_bit is 1
   4. If true append 1 to bin
   5. subtract