

Tender Heart High School

Class: x Computer Application

Teacher: Prabhdeep L

Topic: User Defined Function (Method)

In-built /
Pre defined

user defined function

Eg. Math.Sqrt()

double area(Side)

→ CharAt() → Result

datatype function name

Input
value/argu.

"computer". CharAt(2) → 'm'

Math.Sqrt(16.0) → 4.0

all are predefined in Java

Defining a method

<access specifier> <Return type> <Method name> <Parameter list> ^{datatype with all}

1. Public double area(double l, double b)

```

{
4. double a = l * b;
   return(a);
}

```

1. Header / Method prototype / Function prototype

2. Method Signature

area(l, b)

3. Access specifier

Public

4. Method block

5. Return Statement // last statement will be executed after return

* return statement should always be the last statement.

* only one return statement is executed from block if you add more than one.

```

if (a > b)
return a;
return(a);
else
return(b);

```

// * Method will always return one value only to its caller.

user defined class : Computer Application

L1

Invoking / calling a method

import java.util.*;

Class exam

{

public static double area (double l, double b)

{

double a = l * b;

return (a);

}

Public static void main (String args[])

Scanner sc = new Scanner (System.in);

System.out.println ("Enter length & breadth");

double l = sc.nextDouble();

double b = sc.nextDouble();

exam.ob = new exam(); //X double ar = area(l, b);

double ar = ob.area(l, b); // actual parameters

System.out.println ("The area is" + ar);

if it written no need of creation of object you can call function directly like $\text{double ar} = \text{area}(l, b);$

Formal Parameters

Called Function

calling function

Different ways of defining a function-

1. Receiving value and returning outcome to the caller
2. Receiving value but not returning outcome to the caller
3. Neither Receiving value not returning outcome to the caller

Parameterizing

Not receiving value but returning outcome to the caller

Non-Parameterizing

user defined datatype

Computer Application

L2

Parameterized Methods with return value

Write a class using a function `int num(int)` that accepts a number and finds whether is odd and divisible by 5 or not. It returns 1 if the condition is satisfied otherwise 0. Use a main method to pass the number by value to the function

```
import java.util.*;
```

```
class check
```

```
{
```

```
    int num(int n)
```

```
    {
```

```
        if (n%2 == 1 && n%5 == 0)
```

```
            return 1;
```

```
        else
```

```
            return 0;
```

```
    }
```

```
    public static void main()
```

```
    {
```

```
        Check ob = new check();
```

```
        Scanner sc = new Scanner(System.in);
```

```
        System.out.println("Enter number");
```

```
        int n = sc.nextInt();
```

```
        int ans = ob.num(n);
```

```
        if (ans == 1)
```

// after this it return 1 or 0 and ans store it.

```
            System.out.println("odd number and divisible by 5")
```

```
        else
```

```
            System.out.println("condition not satisfied");
```

```
    }
```

```
}
```

User defined dtype

12

Parameterized Methods // where no return type
with return value is mention then you
have to find out

Write a Prog. to accept a number and check whether the number is palindrome or not by using the function name `rev(int n)`. The function returns the reversed number to the main fn that checks the palindrome number.

```
import java.util.*;
```

```
Class palindrome.
```

```
{
```

```
    public static void main()
```

```
    {
```

```
        Scanner sc = new Scanner(System.in);
```

```
        System.out.println("Enter the number");
```

```
        int n = sc.nextInt();
```

```
        palindrome ob = new palindrome(); // object creation
```

```
        int re = ob.rev(n);
```

```
        if (re == n)
```

```
            System.out.println("Palinno.")
```

```
        else
```

```
            System.out.println("Not palin no.");
```

```
    }
```

```
    int rev(int n)
```

```
    {
```

```
        for(int i = n; i > 0; i = i/10)
```

```
        { int a, r = 0;
```

```
          a = i % 10;
```

```
          r = r * 10 + a;
```

```
        }
```

```
        return(r);
```

```
    }
```

```
}
```

value of n is stored in

User defined datatype

Parameterized functions

// void means fn. is not return any value [No Return type]

Program to input a word and pass this word to a function void disp(String s) that displays all the vowels present in the word.

```
import java.util.*;
```

```
class VOW
```

{ *Shows with not return type*

```
void disp(String s)
```

```
{ char ch;
```

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

```
{ ch = s.charAt(i);
```

```
if (ch == 'A' || ch == 'E' || ch == 'I' || ch == 'O' || ch == 'U');
```

vari. name in called fn. and calling can be same or not

```
{ System.out.print(ch);
```

```
} public static void main(String args[])
```

```
{
```

```
Scanner sc = new Scanner(System.in);
```

```
System.out.println("Enter word");
```

```
String st = sc.next();
```

```
VOW ob = new VOW();
```

```
ob.disp(st); // actual parameter
```

```
}
```

```
}
```

User defined dtype

Parameterized Methods

[NO Return]

Program to input a number. Calculate and display its factorial using function fact(int)

import java.util.*;

class num

{

void fact(int n)

{

long f = 1;

for (int i = 1; i <= n; i++)

{

f = f * i;

}

System.out.println(f);

}

public void main (String args[])

{

Scanner sc = new Scanner (System.in);

num ob = new num();

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

int num1 = sc.nextInt();

ob.fact(num1);

}

}

Sample Input - 4

Output - 24

1x2x3x4=24

Non-Parameterized Functions [without return]

Program to use a function Prime() to input and check whether the entered number is a prime number or not.

```
import java.util.*;
```

```
class PR
```

```
{
```

```
void Prime ()
```

```
{
```

```
Scanner sc = new Scanner (System.in);
```

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

```
int n = sc.nextInt();
```

```
for (int i = 1; i <= n; i++)
```

```
{ if (n % i == 0)
```

```
{ c++;
```

```
if (c == 2)
```

```
System.out.println ("Prime");
```

```
else
```

```
System.out.println ("Non Prime");
```

```
}
```

```
public static void main ()
```

```
{
```

```
PR ob = new PR();
```

```
ob.Prime();
```

```
}
```

Class: X

Computer Application

user defined datatype

Teacher: Prabhdeep Kaur 14

Program to use a function Prime() to input and check whether the entered number is a prime number or not. If the number is prime the function returns 1 else 0 to the main function.

```
import java.util.*;  
class pr  
{  
    int Prime()  
    {  
        Scanner sc = new Scanner(System.in);  
        int c = 0;  
        System.out.println("Enter number");  
        int n = sc.nextInt();  
        for (int i = 1; i <= n; i++)  
        {  
            if (n % i == 0)  
            {  
                c++;  
            }  
        }  
        if (c == 2)  
        {  
            return 1;  
        }  
        else  
        {  
            return 0;  
        }  
    }  
    public static void main()  
    {  
        pr ob = new pr();  
        int a = ob.Prime();  
        if (a == 1)  
        {  
            System.out.print("Prime");  
        }  
        else  
        {  
            System.out.print("Non Prime");  
        }  
    }  
}
```

Class x

Computer Application

User defined datatype

Prabhdeepkaur

User Defined Function

Pass by value/
call by value

Pass by reference/
call by reference

1. A copy of actual Parameters is passed to the formal parameters

1. The reference (address) of actual Parameters is Passed to formal Parameters

2. Any change made in formal Parameters does not reflect on actual parameters

2. Any change made in the formal parameters will be reflected on the actual parameters

3. All the primitive type data like int, float, char etc. are by default passed by value

3. The non-primitive type data like arrays and objects are passed by reference.

Eg. 4.

Class eg

```
void sample(int x, int y)
{
```

```
    x = x + 10
```

```
    y = y + 10
```

```
    System.out.println(x + ' ' + y);
```

```
}
```

```
public static void main ()
```

```
{
```

```
    eg ob. new eg();
```

Tender Heart High School

class: X Computer Application

User defined datatype

Prabdeep

```

int p = 2, q = 3;
ob.sample(p, q);
System.out.println(p + ' ' + q);
}

```

4. Eg.

```

class eg
{
    void sample(int x[])
    {
        for(int i = 0; i < x.length; i++)
        {
            x[i] = x[i] + 10;
        }
        System.out.println("Array");
        for(int i = 0; i < x.length; i++)
            System.out.println(x[i]);
    }
    public static void main()
    {
        int y[] = {1, 3, 5, 7};
        eg ob = new eg();
        ob.sample(y);
        System.out.println("Array");
        for(int i = 0; i < y.length; i++)
            System.out.println(y[i]);
    }
}

```

Tender Heart High School

userdefinedtype

Computer Application

|| In your syllabus program are only on call by value

|| WAP to input a number and check whether it is Armstrong or not using a function `int arms(int)` that returns 1, if the number is armstrong otherwise 0. write the main function to display the result.

Sample Input 153 $[1^3 + 5^3 + 3^3 = 1 + 125 + 27 = 153]$

output It is Armstrong number

```
import java.util.*;
class sample
{
    int arms(int n)
    {
        int s=0, a;
        for(int i=n; i>0; i=i/10)
        {
            a = i%10;
            s = s + (a*a*a);
        }
        if (s == n)
            return 1;
        else
            return 0;
    }
    public static void main ( )
    {
        Scanner sc = new Scanner (System.in);
        sample ob = new sample();
```

Tender Heart High School /
Computer Application

```
System.out.println("Enter the number");  
int n = sc.nextInt();  
int ans = ob.armstrong(n);  
if (ans == 1)  
    System.out.println("It is Armstrong number");  
else  
    System.out.println("Not Armstrong number");
```

```
}  
}
```

Armstrong number logic

$$153 \% 10 = 3$$

$$153 / 10 = 15$$

$$15 \% 10 = 5$$

$$15 / 10 = 1$$

$$1 \% 10 = 1$$

Tender Heart High School

Computer Application

WAP using function fact(int n) to find and display the factor of a number n. Include the main function to input the value of n and call the function fact(int n).

Sample Input is: 6

$6 \% 1 = 0$	$6 \% 3 = 0$	$6 \% 5 = 1$
$6 \% 2 = 0$	$6 \% 4 = 2$	$6 \% 6 = 0$

Output is: 1, 2, 3, 6

```
import java.util.*;
class Factor
{
    public static void main()
    {
        Scanner sc = new Scanner(System.in);
        Factor ob = new Factor();
        System.out.println("Enter the number:");
        int n = sc.nextInt();
        ob.factor(n);
    }
    void factor(int n)
    {
        for (int i = 1; i <= n; i++)
        {
            if (n % i == 0)
                System.out.println(i);
        }
    }
}
```

Tender Heart High School

14

// Statement input of function

Give the prototype

1. A function find which recover a sentence sent and a word wd and returns 1 or 0

int find(String sent, String wd) // no semicolon

2. A function check which receives a character ch and an integer num and return true or false

boolean check(char ch, int num) // don't put ;

3. A function total which receives a sentence and returns its length

int total(String st) // don't put ;

4. A function display which receives an integer n and does not return any value to its caller module.

void display(int n)

5. A function stud which receives one boolean b and one integer n and returns an integer value.

int stud(boolean b, int n)

6. A function apple which does not receive any value but return an integer

int apple()