FACE

PRINTING

- There are three standard streams, all are managed by the java.lang.System class
- **Standard input--referenced by System.in**
  - Used for program input, typically reads input entered by the user.
- **Standard output--referenced by System.out**
  - Used for program output, typically displays information to the user.
- **Standard error--referenced by System.err**
  - Used to display error messages to the user.

PRINTING

FACE

The basic output statement is :

System.out.println( );

Others methods:

1. System.out.println()
2. System.out.print()
3. System.out.printf()

Let us see an example

FACE

```
class AssignmentOperator
{
        public static void main(String[] args)
         {
                System.out.println("Java programming.");
        }
 }
```

Difference between print(), println() and printf()

- print() - prints string inside the quotes.

- println() - prints string inside the quotes similar like print() method. Then the cursor moves to the beginning of the next line.

- printf() - it provides string formatting.

Guess the output

FACE

```
class Output
{
        public static void main(String[] args)
        {
                System.out.println("1. println ");
                System.out.println("2. println ");
                System.out.print("1. print ");
                System.out.print("2. print");
        }
}
```

PRINTING VARIABLES AND LITERALS

To display integers, variables and so on, do not use quotation marks.

```
class Variables
{
        public static void main(String[] args)
        {
                Double number = -10.6;
                System.out.println(5);
                System.out.println(number);
        }
}
```

Print concatenated strings

FACE

You can use + operator to concatenate strings and print it.

```
class PrintVariables
{
        public static void main(String[] args)
        {
                Double number = -10.6;
                System.out.println("I am " + "awesome.");
                System.out.println("Number = " + number);
        }
 }
```

Consider this code snippet

```
int a = 3;
int b = 4;
System.out.println( a + b );
System.out.println( "3" + "4" );
System.out.println( "" + a + b );
System.out.println( 3 + 4 + a + " " + b + a );
System.out.println( "Result: " + a + b );
System.out.println( "Result: " + ( a + b ) );
```

Printing characters

FACE

You can use + operator to concatenate strings and print it.

```
char a=65;
char b='A';
System.out.println(a);
System.out.println(b);
```

FACE

READING INPUT

READING INPUT FROM CONSOLE

FACE

In Java, there are three different ways for reading input from the user in the command line environment(console).

1. Using Buffered Reader Class
2. Using Scanner Class
3. Using Console Class

BUFFERED READER CLASS

FACE

This method is used by wrapping the System.in (standard input stream) in an InputStreamReader which is wrapped in a BufferedReader, we can read input from the user in the command line.

BUFFERED READER CLASS

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
public class Test
{
   public static void main(String[] args)
throws IOException
   {
//Enter data using BufferReader
      BufferedReader reader =
               new BufferedReader(new
InputStreamReader(System.in));
        // Reading data using readLine
        String name = reader.readLine();

        // Printing the read line
        System.out.println(name);
    }
}
```

SCANNER CLASS

FACE

- This is probably the most preferred method to take input.

- The main purpose of the Scanner class is to parse primitive types and strings using regular expressions, however it is also can be used to read input from the user in the command line.

SCANNER CLASS

FACE

```
import java.util.Scanner;

class GetInputFromUser
{
   public static void main(String args[])
   {
      // Using Scanner for Getting Input
from User
      Scanner in = new
Scanner(System.in);
      String s = in.nextLine();
      System.out.println("You entered
string "+s);
      int a = in.nextInt();
      System.out.println("You entered
integer "+a);

      float b = in.nextFloat();
      System.out.println("You entered
float "+b);
   }
}
```

CONSOLE CLASS

FACE

- It has been becoming a preferred way for reading user's input from the command line.

- In addition, it can be used for reading password-like input without echoing the characters entered by the user; the format string syntax can also be used (like System.out.printf()).

CONSOLE CLASS

FACE

```
public class Sample
{
   public static void main(String[] args)
   {
      // Using Console to input data from user
      String name = System.console().readLine();

      System.out.println(name);
   }
}
```

FACE

COMMAND LINE ARGUMENTS

COMMAND LINE ARGUMENTS

The java command-line argument is an argument i.e. passed at run time.

The arguments passed from the console can be received in the java program and it can be used as an input.

Let us see an example

FACE

```
class CommandLineExample
{
        public static void main(String args[])
        {
                System.out.println("Your first argument is: "+args[0]);
        }
}
```

Program 1 : Adding two integers using command line arguments

```
class A
{
        public static void main(String args[])
        {
          System.out.println(args[0]+args[1]);
        }
}
```

Predict the output .

9 

18 

Program 1 : Adding two integers using command line arguments

```
class A{
public static void main(String args[])
{
  System.out.println(Integer.parseInt(args[0]
)+Integer.parseInt(args[1]));
  }
}
```

Predict the output .

9 

Program 2 : Concatenating two strings

```
class A
{
    public static void main(String args[])
    {
        System.out.println(args[0]+args[1]);
    }
}
```

Predict the output .

Hai Hello

Program 3 : Find average of your marks (5 subjects)

```
class A
{
    public static void main(String args[])
    {
      float avg;
      avg = (args[0]+args[1]+args[2]+args[3]+args[4])/5;
      System.out.println(avg);
    }
}
```

Predict the output .

Input : java A 67 98 91 78 98

86.4 ✗

Program 3 : Find average of your marks (5 subjects)

```
class A
{
public static void main(String args[])
{
  float avg;
  avg=
(Float.valueOf(args[0])+Float.valueOf(args[1])+
Float.valueOf(args[2])+Float.valueOf(args[3])+
Float.valueOf(args[4]))/5;
System.out.println(avg);
  }
}
```

Predict the output .

Input : java A 67 98 91 78 98

86.4 

FACE

THANK YOU