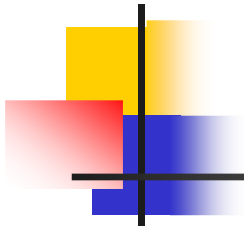
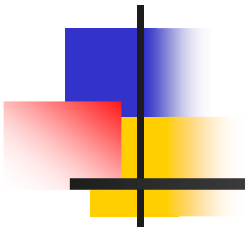
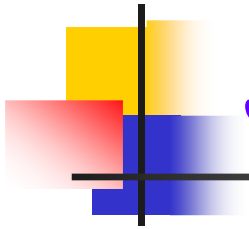


Lecture-2

- Functions
- OOP concepts of Java
  - class and object
- Strings
- Arrays
  - 1D and 2D arrays



Functions



Java functions

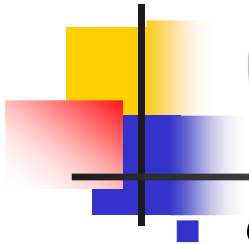
- A group of statements
  - To perform a task
  - Possibly return a value
- Unlike C, functions are part of a class in Java

- Syntax

```
<return_type> fn_name (arguments)
    {
        // function body
    }
```

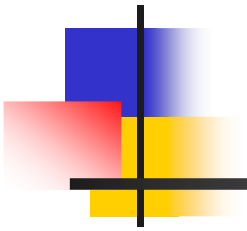
Java functions … example

```
import java.io.*;

public class anyClass                          // Define a class first
{
   int square (int x)                          // function to compute square
     {
        return (x * x);
     }

   public static void main(String args[ ])  //Starting point of the program
     {
        int i = 5;
        anyClass ac = new anyClass();          // Create an object first
        int i_sq = ac.square (5);              // Call its function
        System.out.println ("Square of 5 is: " + i_sq);
     }
}
```

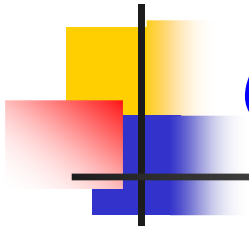


Useful Java functions

- clone ( )
  - Create a copy of an existing object
- equals ( )
  - Checks if two objects are the same
  - This is not quite the same as == operator
- finalize ( )
  - Called to clean up object's resources.
- getClass ( )
  - Returns a class object
- hashCode ( )
  - Returns object's memory address in hexadecimal
- toString ( )
  - Returns a string representation of the object.

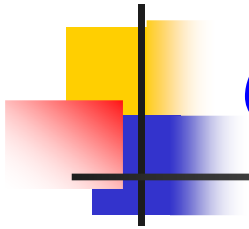


OOP - Class and objects



Class and objects

- class – the basic unit of OOP in Java
- A class typically corresponds to some meaningful entity.
- class has both data and methods.
- Attributes and methods are members of a class
- An instance of a class is an object.
- A class uses methods to interact with other classes/functions.



Class and objects … contd.

- Classes may have both
  - Data attributes
    - Can be of basic or user defined data types.
    - Need to be initialized – typically done in a constructor
  - Methods, Functions
    - Functions that are part of classes
    - Typically interfaces to interact with other classes and functions.
    - Provide APIs to external world to access and manipulate data attributes.

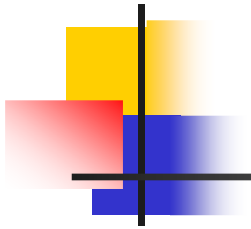
Constructor and destructor … contd.

Constructor

- A function with the same name as the class
- Called when an object is created
- A class can have more than one constructor

Destructor

- There is NO destructor in Java, equivalent to C++ destructor
- In C++
  - Destructor: A function with the name ~classname( )
  - Called when the object goes out of scope, or deleted.
- In Java, closest equivalent is finalize( ) function
  - Used to clean up system resources
    - E.g. close open files, open sockets
    - Clear screen for GUI/graphics objects.
  - Called by system garbage collectors and other resource cleanup functions.



A simple "account" example

```
public class Account
{
  private  int user_SSN;                                  // attribute (data)
  private  int accountNumber;                             // attribute (data)
  public   void withdrawMoney (int amount) { .. } ;       // method
  public   void depositMoney (int amount)   { .. };       // method
  public   void computeInterest( )          { .. };       // method

  public Account() { }                                    // Constructor
  public static void main (String args[ ])                // main function
    {
       Account a = new Account( );                        // Create a new object
       System.out.println ("In account main");
    }
};
```

Account example … contd.

```
import java.io.*;

public class Account
{
  private  int user_SSN;              // attribute (data)
  private  int accountNumber;         // attribute (data)

  // Account constructor
  public Account( )
    {
       user_SSN = -1; accountNumber = -1; // default values
    }

  // Account finalize function
  protected void finalize( )
    {
       System.out.println ("In Account finalize function");
    }

  // Account main function
  public static void main (String args[ ])
    {
       Account a = new Account( );
       System.runFinalizersOnExit(true); // deprecated
    }
};
```

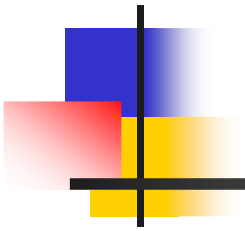
Class definition

Constructor

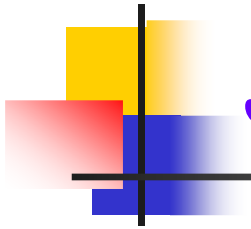
finalize function

main function

Calls finalize functions

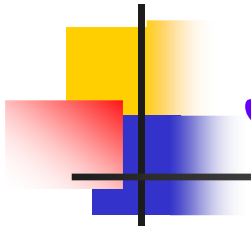


Strings



Java String

- Java String provides a rich collection of functions for
  - Comparison to other strings/sub-strings
  - Character insertion, retrieval
  - Replacing regular expressions
  - Conversion to upper/lower case



Java String

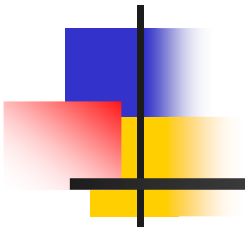
- Java String provides a rich collection of functions for
  - Comparison to other strings/sub-strings
  - Character insertion, retrieval
  - Replacing regular expressions
  - Conversion to upper/lower case

Java String … contd.

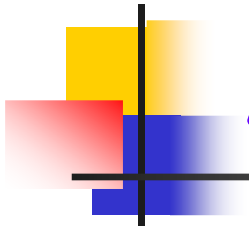
```
public final class String
{
    public int length( )
    public char charAt(int index)
    public void getChars(int srcBegin, int srcEnd, char dst[], int dstBegin)
    public boolean equals(Object anObject)
    public boolean equalsIgnoreCase(String anotherString)
    public int compareTo(String anotherString)
    public int compareToIgnoreCase(String str)
    public int hashCode( )
    public int indexOf(int ch)
    public String replace(char oldChar, char newChar)
    public boolean matches(String regex)
    public String replaceFirst(String regex, String replacement)
    public String replaceAll(String regex, String replacement)
    public String toLowerCase(Locale locale)
    public String toLowerCase( )
    public String toUpperCase( )
    public String toString( )
    public char[ ] toCharArray( )
    …
}
```

Java String … contd.

```
public final class String
{
    public int length( )
    public char charAt(int index)
    public void getChars(int srcBegin, int srcEnd, char dst[], int dstBegin)
    public boolean equals(Object anObject)
    public boolean equalsIgnoreCase(String anotherString)
    public int compareTo(String anotherString)
    public int compareToIgnoreCase(String str)
    public int hashCode( )
    public int indexOf(int ch)
    public String replace(char oldChar, char newChar)
    public boolean matches(String regex)
    public String replaceFirst(String regex, String replacement)
    public String replaceAll(String regex, String replacement)
    public String toLowerCase(Locale locale)
    public String toLowerCase( )
    public String toUpperCase( )
    public String toUpperCase( )
    public String toString( )
    public char[ ] toCharArray( )
    …
}
```



Arrays



Arrays

- Arrays – an ordered sequence of elements of the same type.
- E.g.-1: arr1

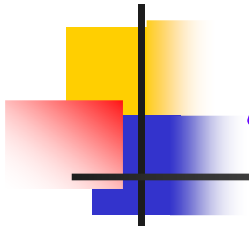
2	4	6	8	10	12	14	16

- E.g.-2: arr2

5	10	15
20	25	30

- arr1[0] = 2; arr[1] = 4 …
- arr2[0][0] = 5; arr2[0][1] = 10; arr2[0][2] = 15;

  arr2[1][0] = 20; arr2[1][1] = 25; arr2[1][2] = 30;



Arrays … contd.

- int [ ] intArray1 = {2, 4, 6, 8, 10};
- float[ ] floatArray1 = {1.1, 2.2, 3.3};
- string strArray1 [ ] = { "abc", "pqr", "xyz" };

- int [ ] intArray2 = new int[10];
  - intArray2[0] = 1; intArray2 [1] = 2;
- double floatArray2 = new double [10];
  - doubleArray2[0] = 1.1; doubleArray2[1] = 2.2;
- String strArray2 = new String[5];
  - strArray2[0] = "abc"; strArray2[1] = "xyz";