Practical Assignment

2021

Name: Sushanta Banerjee

Roll no.: 3490

Registration No.:’1072011400184

Contents

Use native find feature to go to the location with the tag provided.

[P-019] Applet:color fore/back ground

[P-020] Multiple Images in Applet

[P-021] Moving an Image in Applet

[P-022] Drawing Primitive Objects Applet

Main() without String arguments

[P-001]

Code->

```
public class NoArguments
{
    public static void main()
    {
        // Here no string argument is provided
        // The program will compile but give out main not found
        // error during run time
        System.out.println("This line will never see the light of the day.");
    }
}
```

Output->

```
$java NoArguments
Error: Main method not found in class NoArguments, please define the main method as:
   public static void main(String[] args)
or a JavaFX application class must extend javafx.application.Application
```

Run/Compile time Exception

[P-002]

[P-002a] Run time Exception

Code->

```
public class RunTimeExp
{
   public static void main(String[] args)
   {
      int a = 6, b = 0;
      try
      {
         int c = a / b;
         System.out.println("Value of c is " + c);
      }
      catch(ArithmeticException e)
      {
        if (b == 0)
           {
              System.out.println("If you try to divide a number by zero your math teacher will be upset.");
           }
        }
     }
}
```

Output->

```
$java RunTimeExp
If you try to divide a number by zero your math teacher will be upset.
```

[P-002b] Compile time Exception

Code->

```
import java.io.FileInputStream;
import java.io.File;
import java.io.FileNotFoundException;

public class CompileTimeExp
{
   public static void main(String[] args)
   {
      try
      {
         File file = new File("A_file_that_doesnt_exist");
         FileInputStream stream = new FileInputStream(file);
      }
      catch(FileNotFoundException e)
      {
         System.out.println("As long as the file doesn't exist compiler is going to be upset.");
      }
   }
}
```

Output->

```
$java CompileTimeExp
As long as the file doesn't exist compiler is going to be upset.
```

Arithmetic Exception

[P-003]

Code->

```
import java.math.BigDecimal;

public class ArithmeticExp
{
    public static void main(String[] args)
    {
        BigDecimal num1 = new BigDecimal(1);
        BigDecimal num2 = new BigDecimal(3);
        try
        {
            num1 = num1.divide(num2);
        }
        catch(ArithmeticException e)
        {
            System.out.println(e);
        }
    }
}
```

Output->

```
$java ArithmeticExp
java.lang.ArithmeticException: Non-terminating decimal expansion; no exact representable
decimal result.
```

Try, Catch and Finally!

[P-004]

Code->

```
import java.lang.ArithmeticException;
import java.lang.ArrayIndexOutOfBoundsException;

public class tryCF
{
    public static void main(String[] args)
    {
        boolean iAmStupid = true;
        try
        {
            if(iAmStupid) throw new ArithmeticException("Useless Exception 1");
            if(iAmStupid) throw new ArrayIndexOutOfBoundsException("Useless Exception 2");
        }
        catch(ArithmeticException e)
        {
            System.out.println(e);
        }
        catch(ArrayIndexOutOfBoundsException e)
        {
            System.out.println(e);
        }
        finally
        {
            System.out.println("Whatever you do this will always get's to show off.");
        }
    }
}
```

Output->

```
$java tryCF
java.lang.ArithmeticException: Useless Exception 1
Whatever you do this will always get's to show off.
```

Handling Arithmetic Exceptions

[P-005]

Code->

```
public class HandlingArithmeticException
{
    public static void main(String[] args)
    {
        int a = 0, b = 2;
        int c = 0;
        try
        {
            c = b / a;
        }
        catch(ArithmeticException e)
        {
            System.out.println("An exception is occured but handled.");
            System.out.println("Please take care of the value of variable c.");
            System.out.println("Value of c: " + c);
            System.out.println(e);
        }
    }
}
```

Output->

```
$ java HandlingArithmeticException
An exception is occured but handled.
Please take care of the value of variable c.
Value of c: 0
java.lang.ArithmeticException: / by zero
```

Handling ArrayIndexOutOfBound!

[P-006]

Code->

```
public class HandlingArrayIndexOutOfBoundsException
{
   public static void main(String[] args)
   {
      int[] arr = new int[5];
      int num = 4;
      int index = -1;
      try
      {
         arr[index] = num;
      }
      catch(ArrayIndexOutOfBoundsException e)
      {
         System.out.println("Illegal location of array has been tried to access.");
         System.out.println("Please have the value of index between 0 and 5.");
         System.out.println("Current value of index: " + index);
         System.out.println(e);
      }
   }
}
```

Output->

```
$ java HandlingArrayIndexOutOfBoundsException
Illegal location of array has been tried to access.
Please have the value of index between 0 and 5.
Current value of index: -1
java.lang.ArrayIndexOutOfBoundsException: -1
```

Multiple error with single Catch

[P-007]

Code->

```
public class MultipleExp
{
   public static void main(String[] args)
   {
      boolean iLikeFunCode = true;
      try
      {
         if (iLikeFunCode)   throw new ArrayIndexOutOfBoundsException("Nothing nothing");
         if (iLikeFunCode)   throw new ArithmeticException("Really nothing");
      }
      catch(ArrayIndexOutOfBoundsException | ArithmeticException ex)
      {
         System.out.println(ex);
      }
   }
}
```

Output->

```
$ java MultipleExp
java.lang.ArrayIndexOutOfBoundsException: Nothing nothing
```

Use of throws

[P-008]

Code->

```
1 public class Throws
2 {
3    public static void main(String[] args) throws ArithmeticException
4    {
5       int a = 0, b = 2;
6       int c = 0;
7       try
8       {
9          c = b / a;
10       }
11       catch(ArithmeticException e)
12       {
13          System.out.println(e);
14       }
15       System.out.println("C: " + c);
16    }
17 }
```

Output->

```
$ java Throws
java.lang.ArithmeticException: / by zero
C: 0
```

User Defined Exception

[P-009]

Code->

```
 class Age
 {
    private int age;
    public void setAge(int num) throws InvalidAgeException
    {
       if (num < 18) throw new InvalidAgeException("Not yet, Your time has not come yet!");
       else if (num < 0) throw new InvalidAgeException("You gotta be joking right? Your age is
negetive?");
        num = age;
     }

    public int getAge()
    {
       return age;
    }
 }

 class InvalidAgeException extends Exception
 {
    InvalidAgeException(String s)
    {
       super(s);
    }
 }

 public class AgeTestDrive
 {
    public static void main(String[] args)
    {
       Age age = new Age();
       try
       {
          age.setAge(5);
       }
       catch(InvalidAgeException e)
       {
          System.out.println(e);
       }
    }
 }
```

Output->

```
$ java AgeTestDrive
InvalidAgeException: Not yet, Your time has not come yet!
```

Printing Main Thread

[P-010]

Code->

```
public class MainThread extends Thread
{
    public static void main(String[] args)
    {
        Thread t = new Thread();
        t = Thread.currentThread();
        System.out.println("Current Thread: " + t.getName());
        System.out.println("The " + t.getName() + " thread has a priority of value " + t.getPriority());
    }
}
```

Output->

```
$ java MainThread
Current Thread: main
The main thread has a priority of value 5
```

User Defined Thread

[P-011]

Code->

```
class MyThread implements Runnable
{
   public void run()
   {
      System.out.println("Running an user defined thread.");
   }
}

public class UserThread
{
   public static void main(String[] args)
   {
      Runnable aThread = new MyThread();
      Thread newThread = new Thread(aThread);
      newThread.start();
   }
}
```

Output->

```
$ java UserThread
Running an user defined thread.
```

Creating Thread using extend

[P-012]

Code->

```
class ExtendThread extends Thread
{
    public void run()
    {
        System.out.println("Creating a thread extending the thread class.");
    }
}
 public class ExtendThreadTestDrive
{
    public static void main(String[] args)
    {
        ExtendThread thread = new ExtendThread();
        thread.start();
        try
        {
            thread.join();
        }
        catch(InterruptedException e)
        {
            System.out.println(e);
        }
        System.out.println("Successfully closed thread " + thread.getName() + ". With priority " +
thread.getPriority());
    }
}
```

Output->

```
$ java ExtendThreadTestDrive
Creating a thread extending the thread class.
Successfully closed thread Thread-0. With priority 5
```

Creating Thread using Implement

[P-013]

Code->

```
class Super
{
    public void uselessMessage()
    {
        System.out.println("A useless message has been printed on screen.");
    }
}

class Base extends Super implements Runnable
{
   public void moreuselessMessage()
   {
      System.out.println("This is yet another useless message!");
   }
   public void run()
   {
      uselessMessage();
      moreuselessMessage();
   }
}

public class ImplementThreadTestDrive
{
   public static void main(String[] args)
   {
      Runnable test = new Base();
      Thread newThread = new Thread(test);
      newThread.start();
   }
}
```

Output->

```
$ java ImplementThreadTestDrive
A useless message has been printed on screen.
This is yet another useless message!
```

MultiThreading using Extend

[P-014]

Code->

```
class uselessThread extends Thread
{
    public void run()
    {
        System.out.println(this.getName() + " is running!");
    }
}

public class MultiThreadExtend
{
   public static void main(String[] args)
   {
       final int MAX = 10;
       for (int i = 0; i < MAX; i++)
       {
          uselessThread thread = new uselessThread();
         thread.start();
       }
   }
}
```

Output->

```
$ java MultiThreadExtend
Thread-2 is running!
Thread-1 is running!
Thread-0 is running!
Thread-3 is running!
Thread-4 is running!
Thread-5 is running!
Thread-6 is running!
Thread-9 is running!
Thread-7 is running!
Thread-8 is running!
```

MultiThreading using Implement

[P-015]

Code->

```
class uselessThread implements Runnable
{
    public void run()
    {
        System.out.println("A Thread is running!");
    }
}

public class MultiThreadImplement
{
   public static void main(String[] args)
   {
       final int MAX = 10;
       for (int i = 0; i < MAX; i++)
       {
           Runnable reference = new uselessThread();
           Thread thread = new Thread(reference);
           thread.start();
           System.out.println("Thread name is " + thread.getName());
       }
   }
}
```

Output->

```
$ java MultiThreadImplement
Thread name is Thread-0
A Thread is running!
A Thread is running!
Thread name is Thread-1
Thread name is Thread-2
A Thread is running!
Thread name is Thread-3
A Thread is running!
Thread name is Thread-4
A Thread is running!
Thread name is Thread-5
A Thread is running!
Thread name is Thread-6
A Thread is running!
Thread name is Thread-7
A Thread is running!
Thread name is Thread-8
A Thread is running!
Thread name is Thread-9
A Thread is running!
```

getName() and setName()

[P-016]

Code->

```
class myThread extends Thread
{
    public void run()
    {
        System.out.println("Running a thread named " + this.getName());
    }
}

public class GetSetName
{
   public static void main(String[] args)
   {
      final int max = 5;
      for (int i = 0; i < max; i++)
      {
         myThread newThread = new myThread();
         newThread.setName("Thread#" + i);
         newThread.start();
      }
   }
}
```

Output->

```
$ java GetSetName
Running a thread named Thread#0
Running a thread named Thread#2
Running a thread named Thread#1
Running a thread named Thread#4
Running a thread named Thread#3
```

getPriority() and setPriority()

[P-017]

Code->

```
class myThread extends Thread
{
   public boolean valid;
   public void run() {System.out.println(this.getName() + " is running.");}
}

public class GetSetPriority
{
   public static void main(String[] args)
  {
      final int max = 5;
      final int maxPriority = 10;
      myThread[] threads = new myThread[max];
      for (int i = 0; i < max; i++)
      {
         int rand = (int) (Math.random() * maxPriority) + 1;
         threads[i] = new myThread();
         threads[i].setPriority(rand);
         threads[i].valid = true;
      }

      for (int i = 0; i < max; i++)
      {
         System.out.println(threads[i].getName() + " has a priority of " + threads[i].getPriority());
         threads[i].start();
      }
   }
}
```

Output->

```
$ java GetSetPriority
Thread-0 has a priority of 5
Thread-1 has a priority of 8
Thread-0 is running.
Thread-2 has a priority of 10
Thread-1 is running.
Thread-3 has a priority of 1
Thread-2 is running.
Thread-4 has a priority of 1
Thread-4 is running.
Thread-3 is running.
```

Applet goodmorning

[P-018]

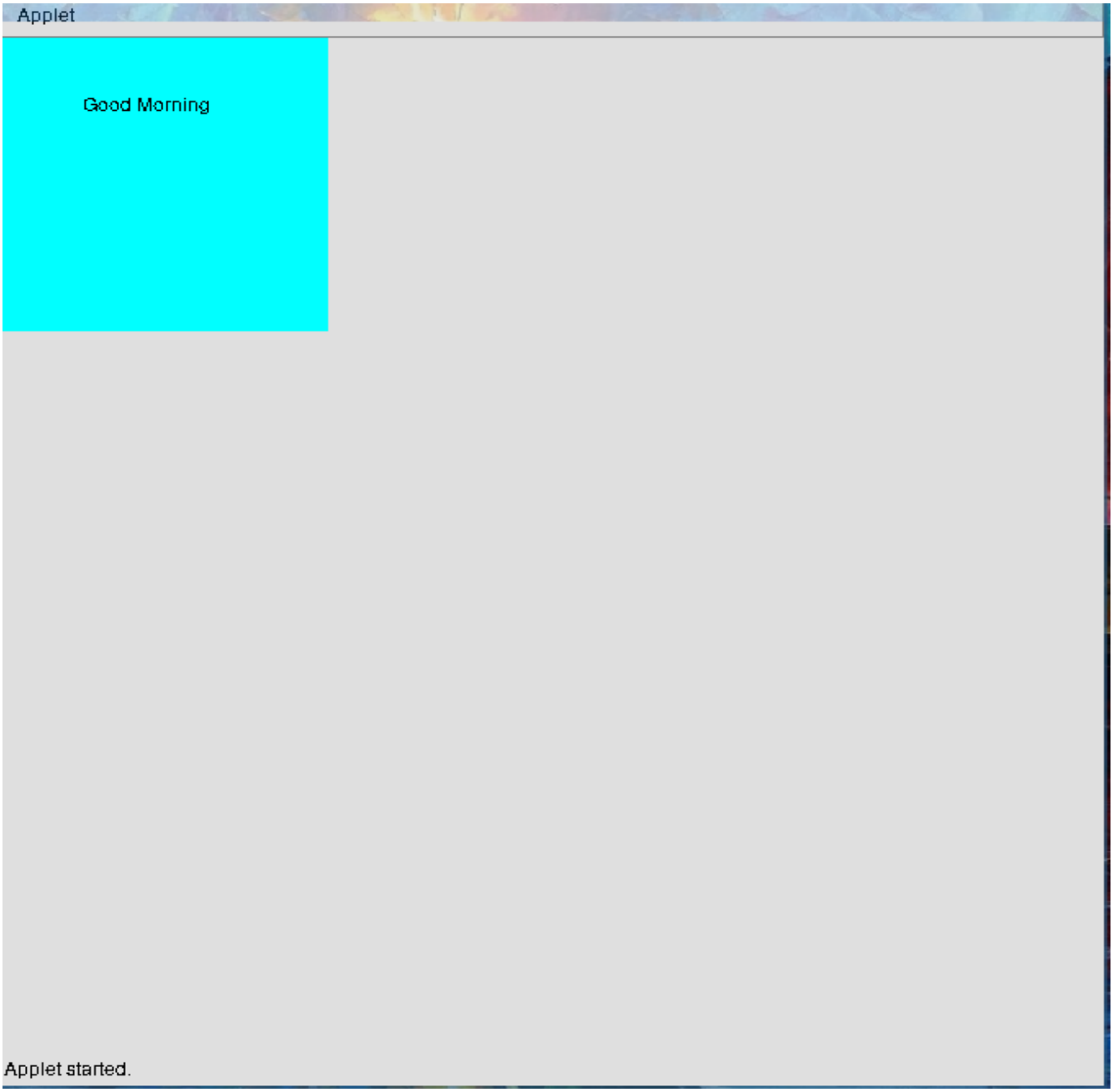
Applet:color fore/back ground

[P-019]

Code->

```
import java.applet.Applet;
import java.awt.Color;
import java.awt.Graphics;

public class SetBackColor extends Applet {

      public void init()
      {
           setBackground(Color.cyan);
          setForeground(Color.red);
      }
     public void paint(Graphics g)
     {
        g.drawString("Good Morning", 50, 50);
     }
}
   /*
    *     <applet code="SetBackColor" width=200 height=200>
    *         </applet>
    *             */
```

Output->



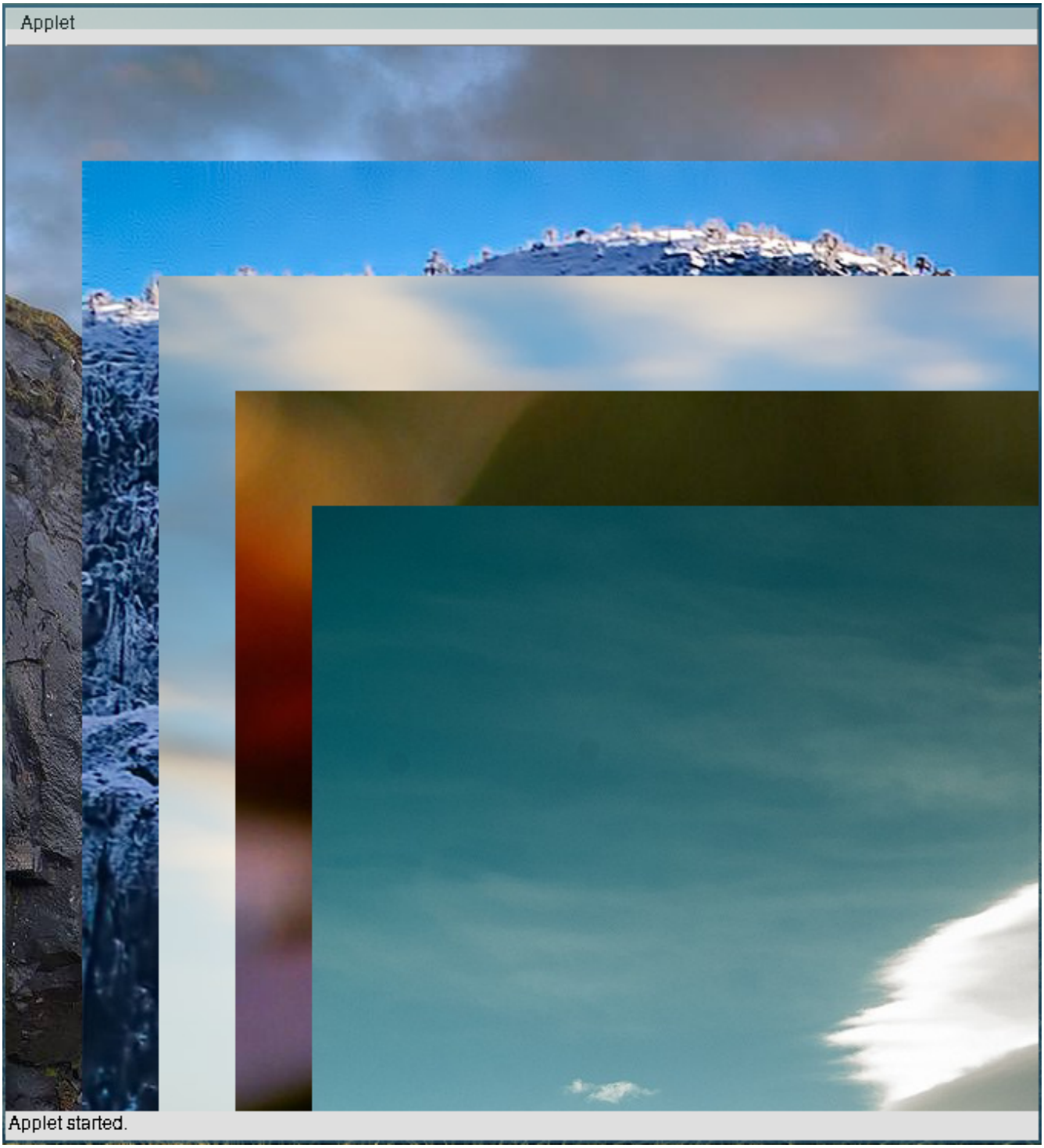
Multiple Images in Applet

[P-020]

Code->

```
import java.awt.*;
import java.applet.*;

public class DisplayImage extends Applet
{
    Image[] picture = new Image[5];

    public void init()
    {
       for (int i = 0; i < 5; i++)
       {
          picture[i] = getImage(getDocumentBase(), "000" + (i+1) + ".jpg");
       }
    }

   public void paint(Graphics g)
   {
       int x = 0, y = 0;
       for (int i = 0; i < 5; i++)
       {
          g.drawImage(picture[i], x, y, this);
          x += 50;
          y += 75;
       }
   }
}

/*
 * <applet code="DisplayImage" width=800 height=600>
 * </applet>
 * */
```

Output->


Applet
Applet started.

Moving an Image in Applet

[P-021]

Code->

```
import java.awt.*;
import java.applet.*;

public class MovingImage extends Applet
{
    Image picture;

    public void init()
    {
       picture = getImage(getDocumentBase(), "bike.gif");
    }

   public void paint(Graphics g)
   {
       for(int i = 0; i < 500; i++)
           g.drawImage(picture, i, 30, this);

       try
       {
           Thread.sleep(100);
       }
       catch(Exception e) {}
   }
}

/*
 * <applet code="MovingImage" width=800 height=600>
 * </applet>
 * */
```

Output->

Drawing Primitive Objects Applet

[P-022]

Code->

```
import java.awt.*;
import java.applet.*;

public class Shapes extends Applet
{
    public void paint(Graphics g)
    {
        g.drawString("Different Shapes", 15, 15);
        g.drawLine(10, 20, 50, 60);
       g.drawRect(10, 70, 40, 40);
       g.setColor(Color.RED);
       g.fillOval(60, 20, 30, 90);
       g.fillArc(60, 135, 80, 40, 180, 180);
       g.fillRoundRect(20, 120, 60, 30, 5, 5);
    }
}

/* <applet code="Shapes" width=200 height=200>
 * </applet>
 */
```

Output->