MVPS's
RAJARSHI SHAHU MAHARAJ POLYTECHNIC, NASHIK-13

COMPUTER TECHNOLOGY DEPARTMENT

ACADEMIC YEAR 2020-21

JAVA PROGRAMING (22412)

MICRO-PROJECT
ON

"Traffic Light Controlling System using Applet and Graphics."

SUBMITTED BY

SR. NO	ENROLLMENT NO	EXAM SEAT NO	STUDENT NAME
1	1610020163		Wani Pushpak Shrikant
2	1910020360		Raut Atharva Satish
3	1910020362		Savant Omkar Vitthal

MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION

CERTIFICATE

This is to certify that Ms. / Mr. Savant Vitthal Omkar
Roll No. 45 of 4th semester of Diploma in **Computer Technology** of Institute **MVPS's Rajarshi Shahu Maharaj Polytechnic, Nashik-13.** **(Code: 1002)** has successfully completed micro-project in **JAVA PROGRAMMING (22412)** for academic year 2020-21 as prescribed in curriculum of MSBTE, Mumbai.

Place: Nashik **Enrollment No. 1910020362**

Date: **Exam Seat No:**

Prof. G. N. Handge	**Prof. P. D. Boraste**	**Dr. D. B. Uphade**
Course Teacher/Guide	**H.O.D**	**Principal**

Seal of institute

MVPS's RAJARSHI SHAHU MAHARAJ POLYTECHNIC,NASIK.

Institute Code: 1002

COMPUTER TECHNOLOGY DEPARTMENT

Log Book for Micro Project

Academic Year : 2020-21 **Scheme: I**
Semester : IV **Class: SYCM**
Name of Course : JAVA PROGRAMMING (JPR) **Course Code: 22412**

Title of the project: Traffic Light Controlling System using Applet and Graphics.

Group Members:

Sr. No.	Roll No.	Enrolment Number	Exam Seat No.	Name of the Student	Signature of student
1	1	1610020163		Wani Pushpak Shrikant	
2	43	1910020360		Raut Atharva Satish	
3	45	1910020362		Savant Omkar Vitthal	

Project Reporting:

Week No.	Discussion& Details	Group Member	Teachers Comment	Sign. Of Course Teacher
1	Discussion on concept of Micro project			
2	Finalization of Micro project topic with proposal submission			
3	Preliminary discussion with guide			
4	Information Gathered			
5	Improvement(Progress) in micro project			
6	Solving Student queries			
7	Rough report writing			
8	Final report writing			
9	Presentation & oral			
10	Final submission			

Name & Signature of Course Teacher

ACKNOWLEDGMENT

I extend our sincere and heartfelt thanks to our esteemed guide, **Prof**. **G. N. Handge** and for his exemplary guidance, monitoring and constant encouragement throughout the semester at crucial junctures and for showing us the right way.

I am grateful to respected coordinator **Prof. G. N. Handge** for permitting me to utilize all the necessary facilities of the Institute.

I would like to extend thanks to our respected Head of the department **Prof. P. D. Boraste** for allowing us to use the facilities available.

ABSTRACT

This paper is scrutinizes the use of different concepts of applets in JAVA programming language, enabling viewer to get the complete concept of different aspects of JAVA programming. Java applets were small applications written in the Java programming language, or another programming language that compiles to Java bytecode, and delivered to users in the form of Java bytecode. To satisfy this we created a simple Java Applet program displaying a Traffic Signal System.

INDEX

DIAGRAM / TABLE

Chapter-1

INTRODUCTION

1. Java:

Java is a high-level, 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. Java applications are typically compiled to bytecode 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 has fewer low-level facilities than either of them. The Java runtime provides dynamic capabilities (such as reflection and runtime code modification) that are typically not available in traditional compiled languages. 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.

Fig.1. Java Programming Language

2. Java Applet:

Java applets were small applications written in the Java programming language, or another programming language that compiles to Java bytecode, and delivered to users in the form of Java bytecode. The user launched the Java applet from a web page, and the applet was then executed within a Java virtual machine (JVM) in a process separate from the web browser itself. A Java applet could appear in a frame of the web page, a new application window, Sun's AppletViewer, or a stand-alone tool for testing applets.

Java applets were introduced in the first version of the Java language, which was released in 1995. Beginning in 2013, major web browsers began to phase out support for the underlying technology applets used to run, with applets becoming completely unable to be run by 2015–2017. Java applets were deprecated since Java 9 in 2017.

Java applets were usually written in Java, but other languages such as Jython, JRuby, Pascal, Scala, or Eiffel (via SmartEiffel) could be used as well.

Java applets run at very fast speeds and until 2011, they were many times faster than JavaScript. Unlike JavaScript, Java applets had access to 3D hardware acceleration, making them well-suited for non-trivial, computation-intensive visualizations. As browsers have gained support for hardware-accelerated graphics thanks to the canvas technology (or specifically WebGL in the case of 3D graphics), as well as just-in-time compiled JavaScript, the speed difference has become less noticeable.

Fig.2. Basic Java Applet

Fig.3. James Gosling - Creator of Java

Fig.4. Sun Microsystems

Chapter-2

ALGORITHM

1. Start
2. Import java.applet
3. Import java.awt
4. Import java awt.event
5. Declare Applet “Signal”
6. Checkbox Statement
    1. If String msg = “Stop”, then go to step
    2. If String msg = “Ready”, then go to step
    3. If String msg = “Go”, then go to step
7. Initiate Class Graphics
8. Execute msg.equals("Stop")()
9. Execute msg.equals("Ready")()
10. Execute msg.equals("Go")()
11. Stop

Chapter-3

FLOWCHART

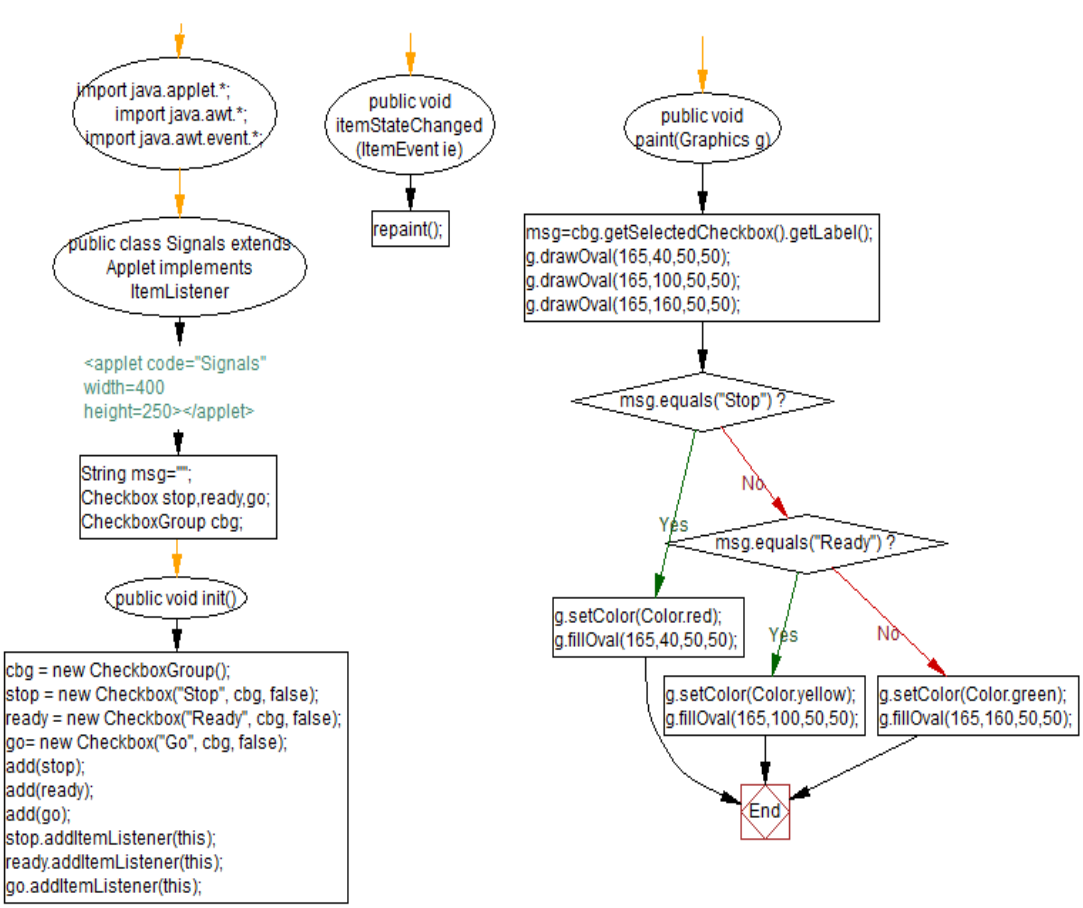

Fig.5. Flowchart

Chapter-4

CODE

```
import java.applet.*;
import java.awt.*;
import java.awt.event.*;
/*<applet code="Signals" width=400 height=250></applet>*/
public class Signals extends Applet implements ItemListener
{
    String msg="";
    Checkbox  stop,ready,go;
    CheckboxGroup cbg;
    public void init()
        {
          cbg = new CheckboxGroup();
          stop = new Checkbox("Stop", cbg, false);
          ready = new Checkbox("Ready", cbg, false);
          go= new Checkbox("Go", cbg, false);
          add(stop);
          add(ready);
          add(go);
          stop.addItemListener(this);
          ready.addItemListener(this);
          go.addItemListener(this);
        }
 public void itemStateChanged(ItemEvent ie)
        {
           repaint();
        }
 public void paint(Graphics g)
        {
         msg=cbg.getSelectedCheckbox().getLabel();
         g.drawOval(165,40,50,50);
         g.drawOval(165,100,50,50);
         g.drawOval(165,160,50,50);
```

```
if(msg.equals("Stop"))
        {
          g.setColor(Color.red);
          g.fillOval(165,40,50,50);
        }
          else if(msg.equals("Ready"))
               {
                g.setColor(Color.yellow);
                g.fillOval(165,100,50,50);
               }
                 else
                      {
                        g.setColor(Color.green);
                        g.fillOval(165,160,50,50);
                       }
         }
}
```

Chapter-5

RESULT

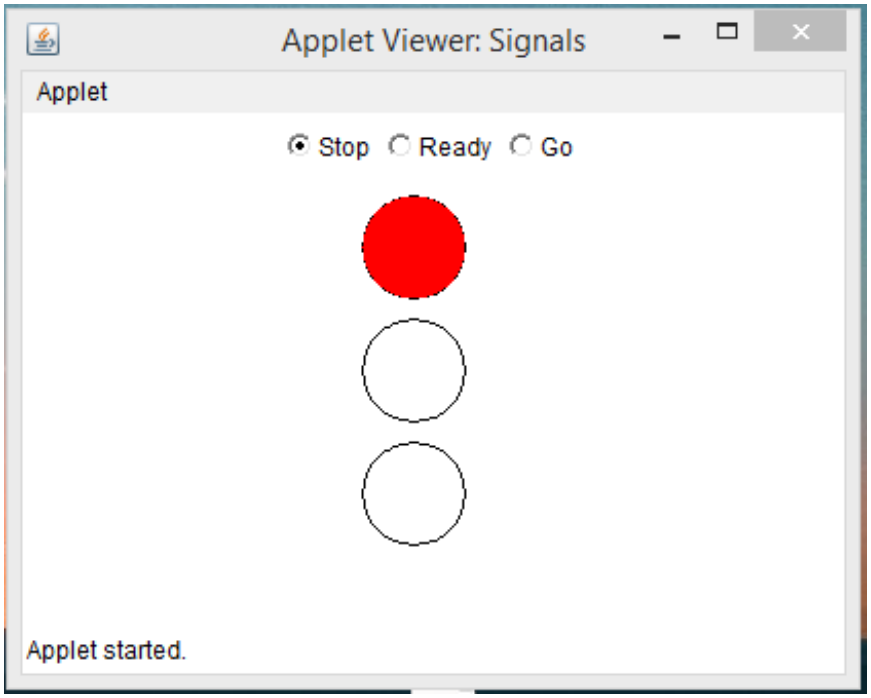

Fig.6. Stop

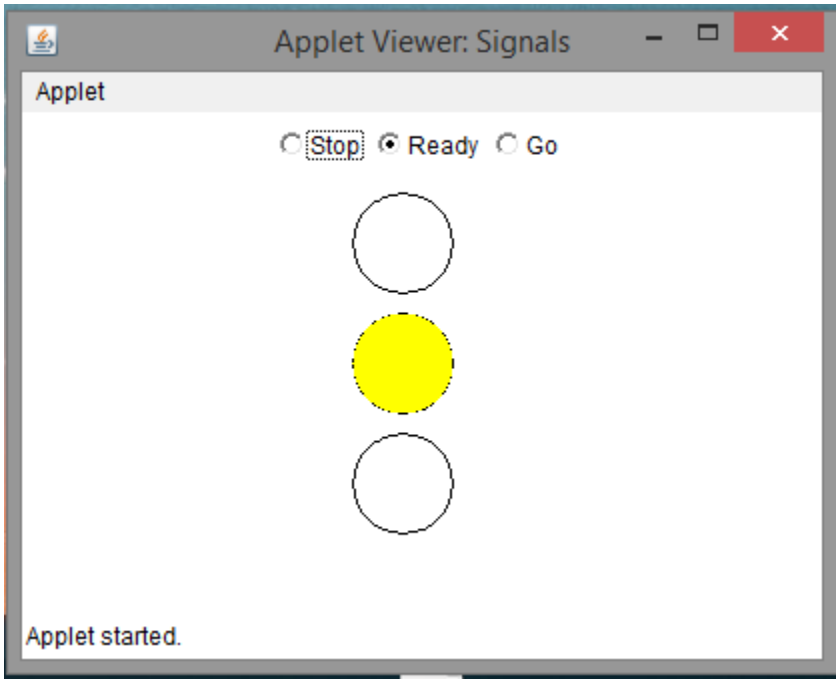

Fig.7. Ready

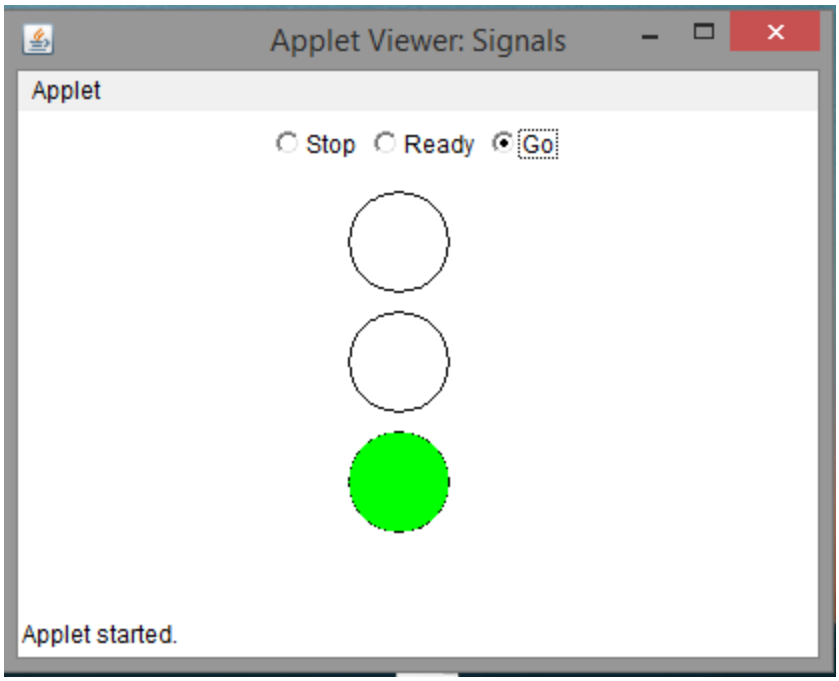

Fig.8. Go

CONCLUSION

Java is a high-level, 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, meaning that compiled Java code can run on all platforms that support Java without the need for recompilation. Programs written in Java have a reputation for being slower and requiring more memory than those written in C++. However, Java programs' execution speed improved significantly with the introduction of just-in-time compilation in 1997/1998 for Java 1.1.

The applets are used to provide interactive features to web applications that cannot be provided by HTML alone. They can capture mouse input and also have controls like buttons or check boxes. In response to user actions, an applet can change the provided graphic content. This makes applets well-suited for demonstration, visualization, and teaching. There are online applet collections for studying various subjects, from physics to heart physiology. As applets were available before HTML5, modern CSS and JavaScript interface DOM were standard, they were also widely used for trivial effects such as mouseover and navigation buttons. This approach, which posed major problems for accessibility and misused system resources, is no longer in use and was strongly discouraged even at the time.

REFERENCES

1. https://en.wikipedia.org/wiki/Java_applet

2. https://en.wikipedia.org/wiki/Java_(programming_language)

3. Chaudhary, Harry H. (July 28, 2014). "Cracking The Java Programming Interview : 2000+ Java Interview Que/Ans"

4. Chan, Rosalie (January 22, 2019). "The 10 most popular programming languages, according to the 'Facebook for programmers'". Business Insider. Archived from the original on June 29, 2019. Retrieved June 29, 2019.

5. In the summer of 1996, Sun was designing the precursor to what is now the event model of the AWT and the JavaBeans component architecture. Borland contributed greatly to this process. We looked very carefully at Delphi Object Pascal and built a working prototype of bound method references in order to understand their interaction with the Java programming language and its APIs.White Paper About Microsoft's Delegates.