PROJECTS

O

PART A

I. TWENTY ONE STICKS GAME

O.01 LEARNING OBJECTIVES

The objectives of the Twenty One Sticks application are as follows:

- To create a simple console-based game in Java
- To use decision making constructs for applying the game's logic

O.02 INTRODUCTION

Twenty One Sticks is a simple console-based gaming application developed in Java. It starts with a collection of twenty one sticks and prompts the user to pick either one or two sticks in one attempt. Based on the user's choice, the computer makes its own choice. Eventually, whosoever picks the last stick loses the game. The user is prompted at the beginning of the game to make a choice whether he wants to play first.

The various variables used in this program are:

- **TwentyOneSticks**: Is the main class which contains the entire code.
- **no_of_sticks**: Is the variable that stores the number of sticks currently available. Initially, it is assigned the value of 21.
- **input_num**: Stores the number of sticks chosen by the user or the computer.

Twenty One Sticks Program

```
import java.util.Scanner; //Scanner class allows reading values of
different types

public class TwentyOneSticks
{
    public status void main(String [] args)
    {
```

```
int no_of_sticks = 21; //The same allows choosing from a total
of 21 sticks
System.out.println(“Do you want to play first? (y/n)”); //User
can either choose to play first or second
Scanner input = new Scanner(System.in);
String first_attempt = input.nextLine();
Scanner read_next = new Scanner(System.in);
int input_num = 0;

while(no_of_sticks>0)
{
    if(first_attempt.equals(“y”) || first_attempt.equals(“Y”))
    {
        System.out.println(“Currently” + no_of_sticks +
        “sticks are available”);
        System.out.println(“Pick your sticks (1 or 2)”);
        input_num = read_next.nextInt();

        if(input_num > 2)
            input_num=2;
        else if(input_num <1)
            input_num=1;

            no_of_sticks = no_of_sticks-input_num;

            if(no_of_sticks <=0)
                System.out.println(“You have lost the
                game.”);
            else
            {
                if(no_of_sticks - 2) % 3 == 0 || (no_of_
                sticks - 2) ==0)
            input_num = 1;
        else
            input_num = 2;
        System.out.println(“Your opponent picks “ + input_num
        + “sticks”);

        no_of_sticks = no_of_sticks - input_num;

        if(no_of_sticks <=0)
            System.out.println(“You have won the game.”);
        }
    }
    else
    {
        if(no_of_sticks -2) % 3 == 0 || (no_of_sticks -2) ==0)
            input_num = 1;
        else
            input_num = 2;
```

```
                    System.out.println(“Your opponent picks “ + input_num
                    + “sticks”);
                    no_of_sticks = no_of_sticks-input_num;

                    if(no_of_sticks <=0)
                        System.out.println(“You have won the game.”);
                    else
                    {
                        system.out.println(Currently “ + no_of_sticks +
                        “sticks are available”);
                        system.out.println(“Pick your sticks (1 or 2)”);
                        input_num = read_next.nextInt();

                        if(input_num>2)
                            input_num = 2;

                        else if(input_num<1)
                            input_num = 1;

                        no_of_sticks = no_of_sticks - input_num;

                        if(no_of_sticks <=0)
                            System.out.println(“You have lost the
                            game.”);
                    }//end of else
                }//end of else
            }//end of while
        }//end of main
    }//end of class
```

O.03 RUNNING THE APPLICATION

To run the Twenty One Sticks application, we need to perform the following steps:

1. Run the following command at the command prompt:

   ```
   javac TwentyOneSticks.java
   ```

2. After successful compilation, run the following command:

   ```
   java TwentyOneSticks
   ```

The output of the program is shown below:

```
Do you want to play first? (y/n) y
Currently 21 sticks are available
Pick your sticks (1 or 2) 2
Your opponent picks 2 sticks
Currently 17 sticks are available
Pick your sticks (1 or 2) 2
Your opponent picks 2 sticks
Currently 13 sticks are available
Pick your sticks (1 or 2) 2
```

```
Your opponent picks 1 sticks
Currently 10 sticks are available
Pick your sticks (1 or 2) 2
Your opponent picks 1 sticks
Currently 7 sticks are available
Pick your sticks (1 or 2) 2
Your opponent picks 1 sticks
Currently 4 sticks are available
Pick your sticks (1 or 2) 2
Your opponent picks 1 sticks
Currently 1 sticks are available
Pick your sticks (1 or 2) 1
You have lost the game.
```

PART B

II. FIND HOST APPLICATION

0.04 LEARNING OBJECTIVES

The objectives of the Find Host application are as follows:

- To find information about a host (Web address)
- To demonstrate the functionality of the various functions of the InetAddress class

0.05 INTRODUCTION

The Find Host application uses the InetAddress class to find information related to the hosts (Web addresses). The InetAddress class uses Domain Name System (DNS) to determine the host information, such as its IP address, host name, etc.

The Find Host application uses the following functions of the InetAddress class:

- **getByName()**: Returns the IP address of the host
- **getAllByName()**: Returns the all the IP addresses of the host
- **getLocalHost()**: Returns the name and IP address of the local host

The application is menu-driven and allows the users to choose the actions that they need to perform.

Find Host Program

```
import java.util.Scanner;
import java.net.*;

class findhost
{
     public static void main(String args[])
     {
```

```
char choice;
String str = “”;
Scanner input = new Scanner(System.in);

while(true)
{
    System.out.println(“Select your choice”);
    System.out.println(“I -> Find IP address for a given host
    name”);
    System.out.println(“A -> Find all IP addresses for a
    given host name”);
    System.out.println(“L -> Find the local host”);
    System.out.println(“E -> Exit”);
    System.out.print(“Your choice: “);
    System.out.flush();

    try
    {
        switch(choice = (char) System.in.read())
        {
            case ‘I’:
            case ‘i’: System.out.println(“Enter host name:
            “);
                str = input.next();
                    try
                    {
                        InetAddress addr=InetAddress.
                        getByName(str);
                        System.out.println(“IP Address of
                        the site is: “+addr);
                    }
                    catch(UnknownHostException e)
                    {
                        System.out.println(“There is an
                        exception: “+e.getMessage());
                    }
                    break;

            case ‘A’:
            case ‘a’: System.out.println(“Enter host name:
            “);
                str = input.next();
                    try
                    {
                        InetAddress[] addrs=InetAddress.
                        getAllByName(str);
                        System.out.println(“IP Addresses of
                        the site are:”);
                        for (int i = 0; i < addrs.length;
                        i++)
                        {
                                System.out.println(addrs[i]);
                            }
                            catch(UnknownHostException e)
                            {
                                System.out.println(“There is an
                                exception: “+e.getMessage());
                            }
                            break;

                    case ‘L’:
                    case ‘l’: try
                        {
                            InetAddress name=InetAddress.
                            getLocalHost();
                            System.out.println(“Local host:”+name.
                            toString());
                        }
                        catch(UnknownHostException e)
                        {
                            System.out.println(“There is an
                            exception: “+e.getMessage());
                        }
                        break;

                    case ‘E’:
                    case ‘e’: System.exit(0);

                    default: System.out.println(“Invalid Choice”);
                }
            }

            catch(Exception e)
            {
                System.out.println(“I/O Error”);
            }
        }//end of while
    }//end of main
}//end of class
```

O.06 RUNNING THE APPLICATION

To run the Find Host application, we need to perform the following steps:

1. Run the following command at the command prompt:

```
javac findhost.java
```

2. After successful compilation, run the following command:

```
java findhost
```

The output of the program is shown below:

```
Select your choice
I -> Find IP address for a given host name
```

```
A -> Find all IP addresses for a given host name
L -> Find the local host
E -> Exit
Your choice: I
Enter host name: www.google.com
IP Address of the site is: 216.239.51.99

Select your choice
I -> Find IP address for a given host name
A -> Find all IP addresses for a given host name
L -> Find the local host
E -> Exit
Your choice: A
Enter host name: www.google.com
IP Addresses of the site are:
www.google.com/72.14.253.147
www.google.com/72.14.253.103
www.google.com/72.14.253.104
www.google.com/72.14.253.99

Select your choice
I -> Find IP address for a given host name
A -> Find all IP addresses for a given host name
L -> Find the local host
E -> Exit
Your choice: L
Local host: default/115.88.3.22

Select your choice
I -> Find IP address for a given host name
A -> Find all IP addresses for a given host name
L -> Find the local host
E -> Exit
Your choice: E
```

Note

The output that you get after running the above program could be slightly different.