

Program Listings: Semester 1

TASK 1: Output the word "Hello" to the Screen:

```
// This programme will display Hello World! on the screen
class Hello {
    public static void main(String args[ ]) {
        System.out.println("Hello World!");
    }
}
```

TASK 2: Create an Applet:

```
/* This is an applet that
   displays "Hello Again World" */
import java.awt.Graphics;
public class HelloApplet extends java.applet.Applet {
    public void paint(Graphics g) {
        g.drawString("Bye Bye World",5,25);
        g.drawRect(2,2,95,55);
        g.drawOval(2,2,95,55);
    }
}
```

TASK 3: Manipulate Person Objects:

```
/**
 *
 * demonstration of the manipulation of "Person" objects
 *
 * Written by: Gareth Evans
 *
 * First Written: 12/June/96
 * Last Rewritten: 22/Oct/98
 *
 */

import java.bangor.*;

public class Chapter3n1
{
    /**
     *
     * main
     *
     */

    public static void main(String[] args)
    {
        // step 1 - declare two "Person" variables
        Person person1 = new Person() ;
        Person person2 = new Person() ;

        // step 2 - set the attributes of "person1"
        person1.setForename("Gareth") ;
        person1.setSurname("Evans") ;
        person1.setAge(18) ;

        // step 3 - set the attributes of "person2"
        person2.setForename("Roy") ;
        person2.setSurname("Evans") ;
        person2.setAge(67) ;

        // step 4 - display the attributes of "person2"
        System.out.print("the second person is ") ;
        System.out.print(person2.getForename()) ;
        System.out.print(" ") ;
        System.out.print(person2.getSurname()) ;
        System.out.print(" (") ;
        System.out.print(person2.getAge()) ;
        System.out.println(")") ;

        // step 5 - leave a blank line
        System.out.println() ;

        // step 6 - display the attributes of "person1"
        System.out.print("the first person is ") ;
        System.out.print(person1.getForename()) ;
        System.out.print(" ") ;
        System.out.print(person1.getSurname()) ;
        System.out.print(" (") ;
        System.out.print(person1.getAge()) ;
        System.out.println(")") ;
    } // end of method main
} // end of class Chapter3n1
```

TASK 4: Manipulate Cards:

```
/**
 *
 * demonstration of the manipulation of Cards
 *
 * Written by Gareth Evans
 *
 * 22/Oct/98
 *
 */

import java.bangor.*;

public class ChooseCard
{
    /**
     *
     * main
     *
     */

    public static void main(String[] args)
    {
        // step 1 - declare two new cards
        Card card1 = new Card();
        Card card2 = new Card();

        // step 2 - display the attributes of card 1
        System.out.print("The first card is the ");
        System.out.print(card1.getValue());
        System.out.print(" of ");
        System.out.print(card1.getSuitString());
        System.out.println();

        System.out.println();

        // step 3 - display attributes of card 2
        System.out.print("The second card is the ");
        System.out.print(card2.getValue());
        System.out.print(" of ");
        System.out.print(card2.getSuitString());
        System.out.println();

    } // end of method main
} // end of class ChooseCard.
```

TASK 5: Manipulate Person Objects (2):

```
/**
 *
 * demonstration of the manipulation of "Person" objects
 *
 * Written by: Gareth Evans
 *
 * First Written: 05/Nov/98
 * Last Rewritten: 05/Nov/98
 *
 */

import java.bangor.*;

public class PersonDisplay
{
    /**
     *
     * main
     *
     */

    public static void main(String[] args)
    {
        // step 1 - declare two "Person" variables
        Person person1 = new Person() ;
        Person person2 = new Person() ;

        // step 1b - declare local variables
        String HumanType = new String();

        // step 2 - set the attributes of "person1"
        person1.setForename("Gareth") ;
        person1.setSurname("Evans") ;
        person1.setAge(16) ;
        person1.setGender(1);

        // step 3 - set the attributes of "person2"
        person2.setForename("Kay") ;
        person2.setSurname("Williams") ;
        person2.setAge(67) ;
        person2.setGender(2);

        // step 4 - display the attributes of "person1"
        System.out.print(person1.getForename() + " " + person1.getSurname()) ;
        System.out.print(" is a " + person1.GetGenderString()) ;

        // step 5 - leave a blank line
        System.out.println() ;

        // step 6 - display the attributes of "person2"
        System.out.print(person2.getForename() + " " + person2.getSurname()) ;
        System.out.print(" is a " + person2.GetGenderString()) ;
        System.out.println();

        // (b) modify the program so that it prints out the name followed by "is a girl" or "is a
        // boy" or "is a woman" or "is a man", as appropriate. For the sake of this exercise, a
        // boy or a girl is a person who is aged 17 or less.

        // step 7 - display age sensitive gender of person 1
        if (person1.getAge() >17)
        {
            if (person1.getGender() < 2)
                {HumanType = "Man";}
            if (person1.getGender() > 1)
                {HumanType = "Woman";}
        }

        if (person1.getAge() <18)
        {
            if (person1.getGender() < 2)
                {HumanType = "Boy";}
            if (person1.getGender() > 1)
                {HumanType = "Girl";}
        }
    }
}
```

```
}

System.out.print(person1.getForename() + " " + person1.getSurname()) ;
System.out.print(" is a " + HumanType) ;
System.out.println();

// step 8 - display age sensitive gender of person 2
if (person2.getAge() >17)
{
    if (person2.getGender() < 2)
        {HumanType = "Man";}
    if (person2.getGender() > 1)
        {HumanType = "Woman";}
}

if (person2.getAge() <18)
{
    if (person2.getGender() < 2)
        {HumanType = "Boy";}
    if (person2.getGender() > 1)
        {HumanType = "Girl";}
}

System.out.print(person2.getForename() + " " + person2.getSurname()) ;
System.out.print(" is a " + HumanType) ;

} // end of method main
} // end of class PersonDisplay
```

TASK 6: Find the Highest Card:

```
/**
 *
 * demonstration of the manipulation of Cards
 *
 * Displays which card has highest value
 *
 * Written by Gareth Evans
 *
 * Started On 5/Nov/98
 *
 * Last Modification On 5/Nov/98
 */

import java.bangor.*;

public class HighCard
{
    /**
     *
     * main
     *
     */

    public static void main(String[] args)
    {
        // step 1 - declare two new cards
        Card card1 = new Card();
        Card card2 = new Card();

        // step 2 - display the attributes of card 1 (COMPUTER)
        System.out.print("The computer's card is the ");
        System.out.print(card1.getValueString());
        System.out.print(" of ");
        System.out.print(card1.getSuitString());
        System.out.println();

        System.out.println();

        // step 3 - display attributes of card 2 (USER)
        System.out.print("The user's card is the ");
        System.out.print(card2.getValueString());
        System.out.print(" of ");
        System.out.print(card2.getSuitString());
        System.out.println();

        // step 4 - show who has highest card value (Not Including Ace)
        if (card2.getValue() > card1.getValue())
        {
            if (card1.getValue() == 0)
            { System.out.println("The Computer Wins!"); }
            else
            System.out.println("The User Wins!");
        }
        if (card1.getValue() > card2.getValue())
        {
            if (card2.getValue() == 0)
            { System.out.println("The User Wins!"); }
            else System.out.println("The Computer Wins!");
        }

        if (card2.getValue() == card1.getValue())
        {
            System.out.println("It is a draw!");
        }

    } // end of method main
} // end of class ChooseCard.
```

TASK 7: Date Time 1:

```
/**
 *
 * Program to output the current Date and Time
 *
 * Written by Gareth Evans
 *
 * 29/Oct/98
 *
 * Write a program to obtain the current date and time,
 * and then display them in the format:
 * The time (sponsored by Java) is 11:47 on 2/20/1997
 *
 */

import java.util.* ;

public class DateTime
{
    /**
     *
     * main
     *
     */

    public static void main(String[] args)
    {
        // step 1 - declare variables

        Date myDate = new Date() ;
        GregorianCalendar myCalendar = new GregorianCalendar() ;
        myCalendar.setTime(myDate) ;

        // step 2 - get values for Date & Time

        int myDayOfMonth = myCalendar.get(Calendar.DAY_OF_MONTH) ;
        int myHourOfDay = myCalendar.get(Calendar.HOUR_OF_DAY) ;
        int myMinute = myCalendar.get(Calendar.MINUTE) ;
        int myMonth = myCalendar.get(Calendar.MONTH) ;
        int myYear = myCalendar.get(Calendar.YEAR) ;

        // step 3 - display Date & Time on screen

        System.out.println();
        System.out.print("The time (sponsored by Java) is ");
        System.out.print(myHourOfDay);
        System.out.print(":");

        if (myMinute < 10)
        {
            System.out.print("0");
            System.out.print(myMinute);
        }

        if (myMinute > 9)
        {
            System.out.print(myMinute);
        }

        System.out.print(" on ");
        myMonth=myMonth+1;
        System.out.print(myMonth);
        System.out.print("/");
        System.out.print(myDayOfMonth);
        System.out.print("/");
        System.out.print(myYear);
        System.out.println();

        // The above code produces the following output:
        // The time (sponsored by Java) is 11:47 on 2/20/1997

    } // end of method main
} // end of class DateTime.
```

TASK 8: Date Time 2:

```
/**
 *
 * Program to output the current Date and Time
 * Updated Version to display the Date and Time in a more
 * "Human Friendly" way.
 *
 * Written by Gareth Evans
 *
 * 12/Nov/98
 *
 * Original Question:
 *
 * Write a program to obtain the current date and time,
 * and then display them in the format:
 * The time (sponsored by Java) is 11:47 on 2/20/1997
 *
 * Updated Question:
 *
 * Display the Date and Time in a more "Human Friendly" way
 */

import java.util.* ;

public class DateTime
{
    /**
     * main
     */

    public static void main(String[] args)
    {
        // step 1 - declare variables

        Date myDate = new Date() ;
        GregorianCalendar myCalendar = new GregorianCalendar() ;
        myCalendar.setTime(myDate) ;

        String daydescription = new String();
        String monthdescription = new String();

        int hournumber ;
        String ampm = new String();

        // step 2 - get values for Date & Time

        int myDayOfMonth = myCalendar.get(Calendar.DAY_OF_MONTH) ;
        int myHourOfDay = myCalendar.get(Calendar.HOUR_OF_DAY) ;
        int myMinute = myCalendar.get(Calendar.MINUTE) ;
        int myMonth = myCalendar.get(Calendar.MONTH) ;
        int myYear = myCalendar.get(Calendar.YEAR) ;

        // step 3 - display Time on screen

        System.out.println();
        System.out.print("The time (sponsored by Java) is ");

        ampm = "AM";
        hournumber = myHourOfDay;

        if (myHourOfDay==13) {hournumber = 1; ampm = "PM";}
        if (myHourOfDay==14) {hournumber = 2; ampm = "PM";}
        if (myHourOfDay==15) {hournumber = 3; ampm = "PM";}
        if (myHourOfDay==16) {hournumber = 4; ampm = "PM";}
        if (myHourOfDay==17) {hournumber = 5; ampm = "PM";}
        if (myHourOfDay==18) {hournumber = 6; ampm = "PM";}
        if (myHourOfDay==19) {hournumber = 7; ampm = "PM";}
        if (myHourOfDay==20) {hournumber = 8; ampm = "PM";}
        if (myHourOfDay==21) {hournumber = 9; ampm = "PM";}
```

```

if (myHourOfDay==22) {hournumber = 10; ampm = "PM";}
if (myHourOfDay==23) {hournumber = 11; ampm = "PM";}

System.out.print(hournumber);
System.out.print(":");

if (myMinute < 10)
{
    System.out.print("0");
    System.out.print(myMinute);
}

if (myMinute > 9)
{
    System.out.print(myMinute);
}

System.out.print(" ");
System.out.print(ampm);

// step 4 - display Date on screen

System.out.print(" on ");

daydescription = "th";
if (myDayOfMonth==1) daydescription = "st";
if (myDayOfMonth==2) daydescription = "nd";
if (myDayOfMonth==3) daydescription = "rd";
if (myDayOfMonth==21) daydescription = "st";
if (myDayOfMonth==22) daydescription = "nd";
if (myDayOfMonth==23) daydescription = "rd";
if (myDayOfMonth==31) daydescription = "st";

System.out.print(myDayOfMonth);
System.out.print(daydescription);
System.out.print(" ");

myMonth=myMonth+1;
if (myMonth==1) monthdescription = "January";
if (myMonth==2) monthdescription = "February";
if (myMonth==3) monthdescription = "March";
if (myMonth==4) monthdescription = "April";
if (myMonth==5) monthdescription = "May";
if (myMonth==6) monthdescription = "June";
if (myMonth==7) monthdescription = "July";
if (myMonth==8) monthdescription = "August";
if (myMonth==9) monthdescription = "September";
if (myMonth==10) monthdescription = "October";
if (myMonth==11) monthdescription = "November";
if (myMonth==12) monthdescription = "December";

System.out.print(monthdescription);
System.out.print(" ");
System.out.print(myYear);
System.out.println();

// The above code produces the following output:
// The time (sponsored by Java) is 11:47 on 2/20/1997

} // end of method main

} // end of class DateTime.

```

TASK 9: Football Program (Assignment 1):

```
/**
 *
 * AWT Demo - "Premier League Champions"
 *
 * Written by: Gareth Evans
 *
 * First Written: 17/Nov/98
 * Last Rewritten: 19/Nov/98
 *
 */

import java.awt.* ;
import java.awt.event.* ;

public class Football extends Frame
    implements WindowListener, ActionListener

{ // start of class Football

    Canvas1 canvas;
    MenuItem start, quitprog, Arsenal, Rovers, ManUtd ;
    Button Ars, BBurn, ManU, quit;
    String champion;

/**
 *
 * constructor
 *
 */

public Football(String ch)

    { // start of constructor method

        champion = ch;

// set up basic window
        setTitle("FA Premier League Champions") ;
        setBackground(Color.green) ;
        setSize(550, 350) ;
        addWindowListener(this) ;

// set up area for face
        canvas = new Canvas1(this) ;
        add("Center", canvas) ;

// set up area with buttons
        Panel p = new Panel() ;
        p.setLayout(new GridLayout(4,1)) ;
        Ars = new Button("Arsenal");
        p.add(Ars);
        Ars.addActionListener(this);
        BBurn = new Button("Blackburn");
        p.add(BBurn);
        BBurn.addActionListener(this);
        ManU = new Button("Manchester Utd");
        p.add(ManU);
        ManU.addActionListener(this);
        quit = new Button("Exit") ;
        p.add(quit) ;
        quit.addActionListener(this) ;
        add("East", p) ;

// set up menu system
        Menu menu = new Menu("File") ;
        Arsenal = new MenuItem("Arsenal") ;
        menu.add(Arsenal) ;
        Arsenal.addActionListener(this) ;
        Rovers = new MenuItem("Blackburn Rovers") ;
        menu.add(Rovers) ;
        Rovers.addActionListener(this) ;
        ManUtd = new MenuItem("Manchester United") ;
        menu.add(ManUtd) ;
        ManUtd.addActionListener(this) ;
        menu.addSeparator();
        quitprog = new MenuItem("Exit") ;
```

```

        menu.add(quitprog) ;
        quitprog.addActionListener(this) ;
MenuBar menuBar = new MenuBar() ;
menuBar.add(menu) ;
setMenuBar(menuBar) ;

    } // end of constructor method

/**
 *
 * main
 *
 */

public static void main(String[] args)

    { // start of main method

        Football f;
        if (args.length == 0)
            f = new Football("Champions");
        else
            f = new Football(args[0]);

        f.setVisible(true) ;

    } // end of main method

/**
 * actionPerformed ocde
 *
 */

public void actionPerformed(ActionEvent event)

    { // start of method actionPerformed

        // deal with "Quit" commands
        if ((event.getSource() == quit) | (event.getSource() == quitprog))
        {
            dispose();
            System.exit(0);
        }

        // deal with "Arsenal" commands
        else if ((event.getSource() == Ars) | (event.getSource() == Arsenal))
        {
            champion = "Arsenal";
            canvas.repaint() ;
        }

        // deal with "Blackburn" commands
        else if ((event.getSource() == BBurn) | (event.getSource() == Rovers))
        {
            champion = "Blackburn";
            canvas.repaint() ;
        }

        // deal with "Manchester United" commands
        else if ((event.getSource() == ManU) | (event.getSource() == ManUtd))
        {
            champion = "Manchester United";
            canvas.repaint() ;
        }

    } // end of method actionPerformed

public void windowClosing(WindowEvent event)

    { // start of method windowClosing

        dispose();
        System.exit(0);

    } // end of method windowClosing

public void windowOpened(WindowEvent event) {}
public void windowIconified(WindowEvent event) {}
public void windowDeiconified(WindowEvent event) {}

```

```

public void windowClosed(WindowEvent event) {}
public void windowActivated(WindowEvent event) {}
public void windowDeactivated(WindowEvent event) {}

} // end of class Football

class Canvas1 extends Canvas

{ // start of class Canvas1

    Football parent ;

/**
 *
 * constructor
 *
 */

public Canvas1(Football f)

{ // start of constructor method

    parent = f;

} // end of constructor method

/**
 *
 * paint
 *
 */

public void paint(Graphics g)

{ // start of paint method

    // set up some dimensions for the drawing
    Dimension dl = getSize() ;
    int cx = dl.width / 2,
    cy = dl.height /2,
    centreCircleRadius = 50,
    arcRadius = 5 ;

// draw the frame
    g.setColor(Color.white) ;
    g.drawRect(10, 10, dl.width - 20, dl.height - 20) ;

// draw the pitch
    g.setColor(Color.white) ;
    g.drawRect(10, dl.height/2-60, 80, 120);
    g.drawRect(dl.width - 90, dl.height/2-60, 80, 120); // 18 yard boxes
    g.drawRect(10, dl.height/2-30, 30, 60);
    g.drawRect(dl.width - 40, dl.height/2-30, 30, 60); // 6 yard boxes
    g.drawLine(dl.width/2, 10, dl.width/2, dl.height - 10); // centre line
    g.drawOval(cx - centreCircleRadius,
              cy - centreCircleRadius,
              centreCircleRadius * 2,
              centreCircleRadius * 2); // centre circle
    g.drawArc(70, dl.height/2-40, 40, 80, -90, 180);
    g.drawArc(dl.width-110, dl.height/2-40, 40, 80, 270, -180); // "D" arcs

// set up screen Fonts
    Font f1 = new Font("TimesRoman", Font.PLAIN, 14) ;
    Font f2 = new Font("TimesRoman", Font.BOLD, 14) ;
    FontMetrics fm1 = g.getFontMetrics(f1) ;
    FontMetrics fm2 = g.getFontMetrics(f2) ;
    g.setColor(Color.black);

// display graphics
    String filenameA = "Prem.gif";
    Image imageA = Toolkit.getDefaultToolkit().getImage(filenameA);
    g.drawImage(imageA, dl.width-145, 11, this);

    if (parent.champion == "Arsenal")

        { // start of Arsenal Section

            // Show Arsenal's text
            String a1 = "Arsenal " ;

```

```

        String a2 = "1997-1998" ;
        int aw1 = fm1.stringWidth(a1) ;
        int aw2 = fm2.stringWidth(a2) ;
        g.setFont(f1) ;
        g.drawString(a1, 20, dl.height-15) ;
        g.setFont(f2) ;
        g.drawString(a2, dl.width -80, dl.height-15) ;

        // Display Arsenal glyph
        String filename1 = "Arsenal.gif";
        Image image1 = Toolkit.getDefaultToolkit().getImage(filename1);
        g.drawImage(image1, 15, dl.height - 70, this);

    } // end of Arsenal Section

else if (parent.champion == "Blackburn")

    { // start of Blackburn Section

        // Show Blackburn's text
        String b1 = "Blackburn Rovers " ;
        String b2 = "1994-1995" ;
        int bw1 = fm1.stringWidth(b1) ;
        int bw2 = fm2.stringWidth(b2) ;
        g.setFont(f1) ;
        g.drawString(b1, 20, dl.height-15) ;
        g.setFont(f2) ;
        g.drawString(b2, dl.width -80, dl.height-15) ;

        // Display Blackburn glyph
        String filename2 = "Blackburn.gif";
        Image image2 = Toolkit.getDefaultToolkit().getImage(filename2);
        g.drawImage(image2, 15, dl.height - 70, this);

    } // end of Blackburn Section

else if (parent.champion == "Manchester United")

    { // start of Manchester Untited Section

        // Show Manchester United's text
        String m1 = "Manchester United " ;
        String m2 = "1992-1993" ;
        String m3 = "1993-1994" ;
        String m4 = "1995-1996" ;
        String m5 = "1996-1997" ;
        int mw1 = fm1.stringWidth(m1) ;
        int mw2 = fm2.stringWidth(m2) ;
        g.setFont(f1) ;
        g.drawString(m1, 20, dl.height-15) ;
        g.setFont(f2) ;
        g.drawString(m2, dl.width -80, dl.height-75) ;
        g.drawString(m3, dl.width -80, dl.height-55) ;
        g.drawString(m4, dl.width -80, dl.height-35) ;
        g.drawString(m5, dl.width -80, dl.height-15) ;

        // Display Manchester United glyph
        String filename3 = "Man Utd.gif";
        Image image3 = Toolkit.getDefaultToolkit().getImage(filename3);
        g.drawImage(image3, 15, dl.height - 70, this);

    } // end of Manchester United Section

else
    { }

// write the general text
String s1 = "FA Premier League " ;
String s2 = "Champions" ;
int w1 = fm1.stringWidth(s1) ;
int w2 = fm2.stringWidth(s2) ;
g.setFont(f1) ;
g.drawString(s1, 20, 30) ;
g.setFont(f2) ;
g.drawString(s2, 20 + w1, 30) ;

    } // end of method paint
} // end of class Canvas1

```

TASK 10: Game 1:

```
/**
 *
 * Exercise 9.1
 *
 * Draws 5 cards; shows total
 *
 * Written by Gareth Evans
 *
 * Started On 26/Nov/98
 *
 * Last Modification On 26/Nov/98
 */

import java.bangor.*;

public class GameOne
{
    /**
     *
     * main
     *
     */

    public static void main(String[] args)
    {
        // step 1 - declare and innitialise variables
        int Total;
        int TempValue;
        Total = 0;

        // step 2 - calculate (and show) the running total
        for (int i = 1; i < 6; i++)
        {
            Card cardtemp = new Card();
            System.out.println();
            TempValue = cardtemp.getValue();
            if (TempValue == 0)
            {
                TempValue = 13;
            }
            System.out.println("The Value of card " + i + " is " + TempValue);
            Total = Total + TempValue;
            System.out.println("The running total after card " + i + " is " + Total);
        }

        System.out.println();

        // step 3 - Is the total under 21?
        if (Total < 22)
        {
            System.out.println("The User Wins! The User has a total of 21 or less!");
        }
        else
        {
            System.out.println("The Computer Wins! The User's Total is more than 21!");
        }

    } // end of method main
} // end of class ChooseCard.
```

TASK 11: Game 2:

```
/**
 *
 * Exercise 9.2
 *
 * Simulates Card Game
 *
 * Written by Gareth Evans
 *
 * Started On 26/Nov/98
 *
 * Last Modification On 26/Nov/98
 */

import java.bangor.*;

public class GameTwo
{
    /**
     *
     * main
     *
     */

    public static void main(String[] args)
    {
        // step 1 - declare and initialise variables
        int TotalCPU;
        int TotalUser;
        int TempValueCPU;
        int TempValueUser;
        int control;
        TotalUser = 0;
        TotalCPU = 0;
        control = 0;

        // step 2 - Play the game!
        while ((TotalUser < 23) & (TotalCPU < 23) & (control < 1))
        {
            for (int i = 1; i < 6; i++)
            {
                control = 1; // stops the loop executing more than once

                Card cardtempCPU = new Card();
                Card cardtempUser = new Card();
                System.out.println();

                TempValueCPU = cardtempCPU.getValue();
                if (TempValueCPU == 0) { TempValueCPU = 13; }

                TempValueUser = cardtempUser.getValue();
                if (TempValueUser == 0) { TempValueUser = 13; }

                System.out.println("The Value of Computer card number " + i + " is " +
                    TempValueCPU);
                TotalCPU = TotalCPU + TempValueCPU;
                System.out.println("The running total after card " + i + " for the computer is " +
                    TotalCPU);
                System.out.println("The Value of User card number " + i + " is " +
                    TempValueUser);
                TotalUser = TotalUser + TempValueUser;
                System.out.println("The running total after card " + i + " for the user is " +
                    TotalUser);
            }
        }

        System.out.println();

        // step 3 - Who is the winner?
        if ((TotalUser < 22) & (TotalCPU > 21))
```

```
{System.out.println("The User Wins! The User has 21 or less; the computer is bust!");}

if ((TotalUser > 21) & (TotalCPU < 22))
{ System.out.println("The Computer Wins! The Computer has 21 or less; the user is
bust!"); }

if ((TotalUser > 21) & (TotalCPU > 21))
{ System.out.println("No one wins! Both players are bust!"); }

if ((TotalUser < 22) & (TotalCPU < 22))
{
    if (TotalUser > TotalCPU)
    { System.out.println("The User Wins. The User's total is less than the Computer's
total"); }

    if (TotalUser < TotalCPU)
    { System.out.println("The Computer Wins. The Computer's total is less than the
user's total"); }

    if (TotalUser == TotalCPU)
    { System.out.println("It is a draw."); }
}

} // end of method main

} // end of class ChooseCard.
```

TASK 12: Game 3:

```
/**
 *
 * Exercise 10.1
 *
 * Simulates Card Game
 * Allows the user to "Stick"
 *
 * Written by Gareth Evans
 *
 * Started On 26/Nov/98
 *
 * Last Modification On 03/Dec/98
 */

import java.bangor.*;

public class GameThree
{
    /**
     *
     * main
     *
     */

    public static void main(String[] args) throws Exception
    {
        // step 1 - declare and initialise variables
        int TotalCPU;
        int TotalUser;
        int TempValueCPU;
        int TempValueUser;
        int control;
        int count;
        String response;
        TotalUser = 0;
        TotalCPU = 0;
        control = 0;
        count = 0;

        // step 2 - Play the game!

        // first part of the game: display information

        System.out.println();
        System.out.println("Welcome to the Blackjack game ");
        System.out.println("You have a maximum of 5 cards");
        System.out.println();

        // second part of the game: deal the cards

        do
        {
            count ++;

            Card cardtempCPU = new Card();
            Card cardtempUser = new Card();
            System.out.println();

            TempValueCPU = cardtempCPU.getValue();
            if (TempValueCPU == 0) { TempValueCPU = 13; }

            TempValueUser = cardtempUser.getValue();
            if (TempValueUser == 0) { TempValueUser = 13; }

            System.out.println("The Value of User card number " + count + " is " +
                TempValueUser);
            TotalUser = TotalUser + TempValueUser;
            System.out.println("The running total after card " +count+ " for the user is " +
                TotalUser);
            System.out.println();
        }
    }
}
```

```

        System.out.println("The Value of Computer card number " + count + " is " +
            TempValueCPU);
        TotalCPU = TotalCPU + TempValueCPU;
        System.out.print("The running total after card " + i);
        System.out.println(" for the computer is " + TotalCPU);
        System.out.println();

        if (count < 5)
        {
            BasicIo.prompt("Deal Again (y or n)? ");
            response = BasicIo.readString().toLowerCase() ;
            System.out.println();
        }
        else {response = "n";}
    }
    while ((response.charAt(0) == 'y') & (count < 5));

    // step 3 - give the computer more cards if 5 cards have not yet been dealt
    //           to the computer

    if (count < 5)
    {
        count++;
        for (int i = count ; i < 6 ; i++)
        {
            if (TotalCPU < 22)
            {
                Card cardtempCPU = new Card();
                TempValueCPU = cardtempCPU.getValue();
                if (TempValueCPU == 0) { TempValueCPU = 13; }
                System.out.println("The Value of Computer card number " + i + " is " +
                    TempValueCPU);
                TotalCPU = TotalCPU + TempValueCPU;
                System.out.print("The running total after card " + i);
                System.out.println(" for the computer is " + TotalCPU);
                System.out.println();
            }
        }
    }

    System.out.println();

    // step 4 - Who is the winner?
    if ((TotalUser < 22) & (TotalCPU > 21))
    {System.out.println("The User Wins! The User has 21 or less; the computer is bust!");}

    if ((TotalUser > 21) & (TotalCPU < 22))
    { System.out.println("The Computer Wins! The Computer has 21 or less; the user is
        bust!"); }

    if ((TotalUser > 21) & (TotalCPU > 21))
    { System.out.println("No one wins! Both players are bust!"); }

    if ((TotalUser < 22) & (TotalCPU < 22))
    {
        if (TotalUser > TotalCPU)
        { System.out.println("The User Wins. The User's total is less than the Computer's
            total"); }

        if (TotalUser < TotalCPU)
        { System.out.println("The Computer Wins. The Computer's total is less than the
            user's total"); }

        if (TotalUser == TotalCPU)
        { System.out.println("It is a draw."); }
    }

} // end of method main

} // end of class ChooseCard.

```

TASK 13: Game 4:

```
/**
 *
 * Exercise 10.2
 *
 * Simulates Card Game
 * Allows the user to "Stick"
 * Plays as many games as the user wants!
 *
 * Written by Gareth Evans
 *
 * Started On 26/Nov/98
 *
 * Last Modification On 03/Dec/98
 */

import java.bangor.*;

public class GameFour
{
    /**
     *
     * main
     *
     */

    public static void main(String[] args) throws Exception
    {
        // step 1 - declare and initialise variables
        int TotalCPU;
        int TotalUser;
        int TempValueCPU;
        int TempValueUser;
        int control;
        int count;
        String response;

        int temp;
        int no_of_games;
        int count_games;
        int wins;
        int draws;
        int losses;
        wins = 0;
        draws = 0;
        losses = 0;

        // ask the user how many games he/she wants to play

        System.out.println();
        BasicIo.prompt("How many blackjack games do you want to play? ");
        no_of_games = BasicIo.readInteger() ;

        temp = no_of_games;
        no_of_games++;
        for (int j = 1 ; j < no_of_games ; j++)
        {
            TotalUser = 0;
            TotalCPU = 0;
            control = 0;
            count = 0;

            // step 2 - Play the game!

            // first part of the game: display information

            System.out.println();
            System.out.println("Welcome to Blackjack game " + j + " out of "+temp);
            System.out.println("You have a maximum of 5 cards");
            System.out.println();

            // second part of the game: deal the cards
```

```

do
{
    count ++;

    Card cardtempCPU = new Card();
    Card cardtempUser = new Card();
    System.out.println();

    TempValueCPU = cardtempCPU.getValue();
    if (TempValueCPU == 0) { TempValueCPU = 13; }

    TempValueUser = cardtempUser.getValue();
    if (TempValueUser == 0) { TempValueUser = 13; }

    System.out.println("The Value of User card number " + count + " is " +
        TempValueUser);
    TotalUser = TotalUser + TempValueUser;
    System.out.println("The running total after card " +count+ " for the user is " +
        TotalUser);
    System.out.println();

    System.out.println("The Value of Computer card number " + count + " is " +
        TempValueCPU);
    TotalCPU = TotalCPU + TempValueCPU;
    System.out.print("The running total after card " + count);
    System.out.println(" for the computer is " + TotalCPU);
    System.out.println();

    if (count < 5)
    {
        BasicIo.prompt("Deal Again (y or n)? ");
        response = BasicIo.readString().toLowerCase() ;
        System.out.println();
    }
    else {response = "n";}
}
while ((response.charAt(0) == 'y') & (count < 5));

// step 3 - give the computer more cards if 5 cards have not yet been dealt
//           to the computer

if (count < 5)
{
    count++;
    for (int i = count ; i < 6 ; i++)
    {
        if (TotalCPU < 22)
        {
            Card cardtempCPU = new Card();
            TempValueCPU = cardtempCPU.getValue();
            if (TempValueCPU == 0) { TempValueCPU = 13; }
            System.out.println("The Value of Computer card number " + i + " is " +
                TempValueCPU);
            TotalCPU = TotalCPU + TempValueCPU;
            System.out.print("The running total after card " + count);
            System.out.println(" for the computer is " + TotalCPU);
            System.out.println();
        }
    }
}

System.out.println();

// step 4 - Who is the winner?
if ((TotalUser < 22) & (TotalCPU > 21))
{ System.out.println("The User Wins! The User has 21 or less; the computer is bust!");
    wins++;}

if ((TotalUser > 21) & (TotalCPU < 22))
{ System.out.println("The Computer Wins! The Computer has 21 or less; the user is
    bust!"); losses++;}

if ((TotalUser > 21) & (TotalCPU > 21))
{ System.out.println("No one wins! Both players are bust!"); draws++;}

if ((TotalUser < 22) & (TotalCPU < 22))
{
    if (TotalUser > TotalCPU)

```

```
{ System.out.println("The User Wins. The User's total is less than the Computer's
    total"); wins++;}

if (TotalUser < TotalCPU)
{ System.out.println("The Computer Wins. The Computer's total is less than the
    user's total"); losses++;}

if (TotalUser == TotalCPU)
{ System.out.println("It is a draw."); draws++;}

}

// display game information
System.out.println();
System.out.println("After Game " + j + ",");
System.out.println();
System.out.println("User Wins: "+ wins);
System.out.println("Draws: "+draws);
System.out.println("User Losses: "+losses);

} // end of all games

    } // end of method main

} // end of class ChooseCard.
```

TASK 14: Game 5:

```
/**
 *
 * Exercise 10.3
 *
 * Simulates Card Game
 * Allows the user to "Stick"
 * Plays as many games as the user wants!
 * Allows the computer to "stick"
 * The computer sticks at 15
 *
 * Written by Gareth Evans
 *
 * Started On 26/Nov/98
 *
 * Last Modification On 03/Dec/98
 */

import java.bangor.*;

public class GameFive
{
    /**
     *
     * main
     *
     */

    public static void main(String[] args) throws Exception
    {
        // step 1 - declare and initialise variables
        int TotalCPU;
        int TotalUser;
        int TempValueCPU;
        int TempValueUser;
        int control;
        int count;
        String response;

        int temp;
        int no_of_games;
        int count_games;
        int wins;
        int draws;
        int losses;
        wins = 0;
        draws = 0;
        losses = 0;

        // ask the user how many games he/she wants to play

        System.out.println();
        BasicIo.prompt("How many blackjack games do you want to play? ");
        no_of_games = BasicIo.readInteger() ;

        temp = no_of_games;
        no_of_games++;
        for (int j = 1 ; j < no_of_games ; j++)
        {
            TotalUser = 0;
            TotalCPU = 0;
            control = 0;
            count = 0;

            // step 2 - Play the game!

            // first part of the game: display information

            System.out.println();
            System.out.println("Welcome to Blackjack game " + j + " out of " + temp);
            System.out.println("You have a maximum of 5 cards");
            System.out.println();
        }
    }
}
```

```

// second part of the game: deal the cards

do
{
    count ++;

    Card cardtempCPU = new Card();
    Card cardtempUser = new Card();
    System.out.println();

    TempValueCPU = cardtempCPU.getValue();
    if (TempValueCPU == 0) { TempValueCPU = 13; }

    TempValueUser = cardtempUser.getValue();
    if (TempValueUser == 0) { TempValueUser = 13; }

    System.out.println("The Value of User card number " + count + " is " +
        TempValueUser);
    TotalUser = TotalUser + TempValueUser;
    System.out.println("The running total after card " +count+ " for the user is " +
        TotalUser);
    System.out.println();

    if (TotalCPU < 16)
    {System.out.println("The Value of Computer card number " + count + " is " +
        TempValueCPU);
    TotalCPU = TotalCPU + TempValueCPU;}

    System.out.print("The running total after card " + count);
    System.out.println(" for the computer is " + TotalCPU);
    System.out.println();

    if (count < 5)
    {
        BasicIo.prompt("Deal Again (y or n)? ");
        response = BasicIo.readString().toLowerCase() ;
        System.out.println();
    }
    else {response = "n";}
}
while ((response.charAt(0) == 'y') & (count < 5));

// step 3 - give the computer more cards if 5 cards have not yet been dealt
//           to the computer

if (count < 5)
{
    count++;
    for (int i = count ; i < 6 ; i++)
    {
        if (TotalCPU < 16)
        {
            Card cardtempCPU = new Card();
            TempValueCPU = cardtempCPU.getValue();
            if (TempValueCPU == 0) { TempValueCPU = 13; }
            System.out.println("The Value of Computer card number " + i + " is " +
                TempValueCPU);
            TotalCPU = TotalCPU + TempValueCPU;
            System.out.print("The running total after card " + count);
            System.out.println(" for the computer is " + TotalCPU);
            System.out.println();
        }
    }
}

System.out.println();

// step 4 - Who is the winner?
if ((TotalUser < 22) & (TotalCPU > 21))
{ System.out.println("The User Wins! The User has 21 or less; the computer is bust!");
    wins++;}

if ((TotalUser > 21) & (TotalCPU < 22))
{ System.out.println("The Computer Wins! The Computer has 21 or less; the user is
    bust!"); losses++;}

if ((TotalUser > 21) & (TotalCPU > 21))

```

```
{ System.out.println("No one wins! Both players are bust!"); draws++;}

if ((TotalUser < 22) & (TotalCPU < 22))
{
    if (TotalUser > TotalCPU)
    { System.out.println("The User Wins. The User's total is less than the Computer's
        total"); wins++;}

    if (TotalUser < TotalCPU)
    { System.out.println("The Computer Wins. The Computer's total is less than the
        user's total"); losses++;}

    if (TotalUser == TotalCPU)
    { System.out.println("It is a draw."); draws++;}
}

// display game information
System.out.println();
System.out.println("After Game " + j + ",");
System.out.println();
System.out.println("User Wins: "+ wins);
System.out.println("Draws: "+draws);
System.out.println("User Losses: "+losses);

} // end of all games

    } // end of method main

} // end of class ChooseCard.
```

TASK 15: Conversion 1:

```
/**
 *
 * Program to convert ounces to grams
 *
 * Written by Gareth Evans
 *
 * 10/Dec/98
 *
 * A program to prompt the user to input an amount in ounces, and return the amount in
 * grams (1 ounce = 28 grams).
 *
 */

import java.util.* ;
import java.bangor.* ;

public class Convert1
{
    /**
     *
     * main
     *
     */

    public static void main(String[] args) throws Exception
    {
        // step 1 - declare variables

        int ounces;

        // step 2 - ask the user for an integer

        System.out.println();
        BasicIo.prompt("Input an integer in ounces to convert to grams ") ;
        ounces = BasicIo.readInteger() ;
        System.out.println();
        System.out.println("Your value, " + ounces + " ounces, is " + ounces*28 + " grams");
    } // end of method main
} // end of class DateTime.
```

TASK 16: Conversion 2:

```
/**
 *
 * Program to convert pints to litres
 *
 * Written by Gareth Evans
 *
 * 10/Dec/98
 *
 * A program so that the user can type in a number of pints, to be converted to liters (1
 * pint = 0.568 liters).
 *
 */

import java.util.* ;
import java.bangor.* ;

public class Convert2
{
    /**
     *
     * main
     *
     */

    public static void main(String[] args) throws Exception
    {
        // step 1 - declare variables

        double pints;

        // step 2 - ask the user for an integer

        System.out.println();
        BasicIo.prompt("Input a number in pints to convert to litres ") ;
        pints = BasicIo.readDouble() ;
        System.out.println();
        System.out.println("Your value, " + pints + " pints, is " + pints*0.568 + " litres");
    } // end of method main
} // end of class DateTime.
```

TASK 17: Conversion 3:

```
/**
 *
 * Program to convert pints to litres
 * and/or convert ounces to grams
 *
 * Written by Gareth Evans
 * 10/Dec/98
 *
 */

import java.util.* ;
import java.bangor.* ;

public class Convert3
{
    /**
     * main
     */

    public static void main(String[] args) throws Exception
    {
        // step 1 - declare variables

        double pints;
        int ounces;
        int choose;
        char again;

        // step 2 - ask the user what conversion he/she wants

        do
        {
            System.out.println();
            BasicIo.prompt("Input 1 to convert from pints to litres; 2 to convert from ounces to
                grams ");
            choose = BasicIo.readInteger();

            // step 3 - execute the appropriate action

            if (choose == 1)
            {
                System.out.println();
                BasicIo.prompt("Input a number in pints to convert to litres ") ;
                pints = BasicIo.readDouble() ;
                System.out.println();
                System.out.print("Your value, " + pints + " pints, is ");
                System.out.println(pints*0.568 + " litres");
                System.out.println();
            }

            if (choose == 2)
            {
                System.out.println();
                BasicIo.prompt("Input an integer in ounces to convert to grams ") ;
                ounces = BasicIo.readInteger() ;
                System.out.println();
                System.out.println("Your value, " + ounces + " ounces, is " + ounces*28 + "
                    grams");
                System.out.println();
            }

            // step 4 - ask the user if he/she wants another conversion

            BasicIo.prompt("Do you want another conversion. (y/n). ");
            choose = BasicIo.readCharacter();
            System.out.println();

        } while (choose == 'y');

    } // end of method main
} // end of class DateTime.
```

TASK 18: The Money Problem:

```
/**
 *
 * The $1, 50¢, 10¢, 1¢ problem.
 *
 * Written by Gareth Evans
 *
 * 10/Dec/98
 */

import java.util.* ;
import java.bangor.* ;

public class Money
{
    /**
     * main
     */

    public static void main(String[] args) throws Exception
    {
        // step 1 - declare variables

        int input;
        int originput;
        int onehundred;
        int fifty;
        int ten;
        int one;

        onehundred=0;
        fifty=0;
        ten=0;
        one = 0;

        System.out.println();
        BasicIo.prompt("Input a number in cents ");
        input = BasicIo.readInteger();
        originput = input;

        do
        {
            if ((input-100)>=-1)
            {
                onehundred++;
                input=input-100;
            }
        }
        while ((input-100)>=-1);

        do
        {
            if ((input-50)>=-1)
            {
                fifty++;
                input=input-50;
            }
        }
        while ((input-50)>=-1);

        do
        {
            if ((input-10)>=-1)
            {
                ten++;
                input=input-10;
            }
        }
        while ((input-10)>=-1);
    }
}
```

```
do
{
    if ((input-1)>-1)
    {
        one++;
        input=input-1;
    }
}
while ((input-1)>-1);

System.out.println();
System.out.println("Your input, " + originput + " cents, can be expressed as...");
System.out.print(onehundred + " * 1 dollar, " + fifty + " * 50 cents, ");
System.out.println(ten + " * 10 cents, and " + one + " * 1 cents");

} // end of method main

} // end of class DateTime.
```

TASK 19: The Pallindromic Problem:

```
/**
 *
 * The Pallindromic problem.
 *
 * Written by Gareth Evans
 *
 * 10/Dec/98
 */

import java.util.* ;
import java.bangor.* ;
import java.beans.* ;

public class Pallindromic
{
    /**
     *
     * main
     *
     */

    public static void main(String[] args) throws Exception
    {
        // step 1 - declare variables

        String input;
        int LengthOfString;
        int pivot;
        int modcheck;

        LengthOfString = 0;

        // step 2 - check if a String is Pallindromic

        System.out.println();
        BasicIo.prompt("Enter a string to be tested if it is Pallindromic ");
        input = BasicIo.readString();

        input = LCase$(input);
        LengthOfString = length(input);
        modcheck = LengthOfString MOD 2;

        modcheck=0;
        LengthOfString=0;

        if (modcheck > 0)
        {
            LengthOfString = (LengthOfString+1);
        }

        pivot = LengthOfString/2;

    } // end of method main
} // end of class DateTime.
```