



Instruction to students:

1. Language allowed to answer is the **English language**.
2. You should attempt **ALL the 34 MCQ** questions in **Section I**.
3. You should attempt **All the 6 Points** of **Section II**.
4. You should attempt **2 out of Section III**.
5. The exam paper is **13 pages long**, and is in **3 sections**.
6. The approximate allocation of **marks** is shown in brackets by the questions.
7. You cannot consult any printed or hand written material, and you cannot talk/communicate with others students.
8. Please check that your Exam Paper is complete.

Section I : <u>MCQ Form</u>							
Questions	1	2	3	4	5	6	7
Answer Key							
Questions	8	9	10	11	12	13	14
Answer Key							
Questions	15	16	17	18	19	20	21
Answer Key							
Questions	22	23	24	25	26	27	28
Answer Key							
Questions	29	30	31	32	33	34	
Answer Key							

Answer the following questions:

Section I

[Total 48]

Q (1): Multiple choice questions Place the answer on the special MCQ form.

1 Analyze the following code: (Choose Two) [2 Marks]

```
public class Test {
    public static void main(String[] args) {
        A a = new A();
        a.print();
    }
}
class A {
    String s;
    A(String s) {
        this.s = s;
    }
    void print() {
        System.out.println(s);
    }
}
```

- A. The program has a compilation error because class A is not a public class.
- B. The program has a compilation error because class A does not have a default constructor.
- C. The program compiles and runs fine and prints nothing.
- D. D. The program would compile and run if you change A a = new A() to A a = new A("5").

2 Analyze the following code. [2 Marks]

```
public class Test {
    public static void main(String[] args) {
        int n = 2;
        xMethod(n);
        System.out.println("n is " + n);
    }
    void xMethod(int n) {
        n++;
    }
}
```

- A. The code has a compile error because xMethod does not return a value.
- B. The code has a compile error because xMethod is not declared static.
- C. The code prints n is 1. D. The code prints n is 2. E. The code prints n is 3.

3 Analyze the following code and choose the best answer: [2 Marks]

```
public class Foo {
    private int x;
    public static void main(String[] args) {
        Foo foo = new Foo();
        System.out.println(foo.x);
    }
}
```

- A. Since x is private, it cannot be accessed from an object foo.
- B. Since x is defined in the class Foo, it can be accessed by any method inside the class without using an object. You can write the code to access x without creating an object such as foo in this code.
- C. Since x is an instance variable, it cannot be directly used inside a main method. However, it can be accessed through an object such as foo in this code.
- D. You cannot create a self-referenced object; that is, foo is created inside the class Foo.

4 Which is the advantage of encapsulation? [1 Mark]

- A. Only public methods are needed. B. Making the class final causes no consequential changes to other code.
- C. It changes the implementation without changing a class's contract and causes no consequential changes to other code.
- D. It changes a class's contract without changing the implementation and causes no consequential changes to other code.

5 What is the output of the following code? [1 Mark]

```
String s = "University";  
s.replace("i", "ABC");  
System.out.println(s);
```

- A. UnABCversity B. UnABCversABCty C. UniversABCty D. University

6 Which of the following is the correct statement to return JAVA? [1 Mark]

- A. toUpperCase("Java") B. "Java".toUpperCase("Java") C. "Java".toUpperCase() D. String.toUpperCase("Java")

7 The following program displays _____? [2 Marks]

```
public class Test {  
    public static void main(String[] args) {  
        String s = "Java";  
        StringBuilder buffer = new StringBuilder(s);  
        change(buffer);  
        System.out.println(buffer); }  
    private static void change(StringBuilder buffer) {  
        buffer.append(" and HTML");  
    } }  
}
```

- A. Java B. Java and HTML C. and HTML D. nothing is displayed

8 Which class do you use to write data into a text file? [1 Mark]

- A. File B. PrintWriter C. Scanner D. System

9 Which of the following are valid methods on the button jbt? (Choose two) [1 Mark]

- A. jbt.setMnemonic("A"); B. jbt.setMnemonic('A'); C. jbt.setIconTextGap(50); D. jbt.setTextGap(50);

10 Which method can be used to read a whole line from the file? [1 Mark]

- A. next B. nextLine C. nextInt D. nextDouble

11 You can declare two variables with the same name in _____. [1 Mark]

- A. a method one as a formal parameter and the other as a local variable B. a block
C. two nested blocks in a method (two nested blocks means one being inside the other) D. different methods in a class

12 Analyze the following code: (Choose two) [2 Marks]

```
class Test {  
    private double i;  
    public Test(double i) {  
        this.t();  
        this.i = i; }  
    public Test() {  
        System.out.println("Default constructor");  
        this(1); }  
    public void t() {  
        System.out.println("Invoking t"); }  
}
```

- A. this.t() may be replaced by t(). B. this.i may be replaced by i.
C. this(1) must be called before System.out.println("Default constructor"). D. this(1) must be replaced by this(1.0).

13 Which of the following can be placed in the blank line in the following code? [1 Mark]

```
public class Test {  
    private int id;  
    public void m1() {  
        _____.id = 45; }  
}
```

- A. this B. Test

14 Analyze the following code: [2 Marks]

```
public class Test {
    public static void main(String[] args) {
        B b = new B();
        b.m(5);
        System.out.println("i is " + b.i);    }
}
class A {
    int i;
    public void m(int i) {
        this.i = i;    }
}
class B extends A {
    public void m(String s) {    }
}
```

- A. The program has a compilation error, because m is overridden with a different signature in B.
- B. The program has a compilation error, because b.m(5) cannot be invoked since the method m(int) is hidden in B.
- C. The program has a runtime error on b.i, because i is not accessible from b.
- D. The method m is not overridden in B. B inherits the method m from A and defines an overloaded method m in B.

15 What is the output of the following code? [2 Marks]

```
public class Test {
    public static void main(String[] args) {
        new Person().printPerson();
        new Student().printPerson();    }
}
class Student extends Person {
    private String getInfo() {
        return "Student";    }
}
class Person {
    private String getInfo() {
        return "Person";    }
    public void printPerson() {
        System.out.println(getInfo());    }
}
```

- A. Person Person
- B. Person Student
- C. Student Student
- D. Student Person

16 Analyze the following code. [2 Marks]

```
// Program 1:
public class Test {
    public static void main(String[] args) {
        Object a1 = new A();
        Object a2 = new A();
        System.out.println(a1.equals(a2));    }
}
class A {
    int x;
    public boolean equals(A a) {
        return this.x == a.x;    }
}
```

```
// Program 2:
public class Test {
    public static void main(String[] args) {
        A a1 = new A();
```

```

A a2 = new A();
System.out.println(a1.equals(a2)); }
}
class A {
int x;
public boolean equals(A a) {
return this.x == a.x; }
}

```

- A. Program 1 displays true and Program 2 displays true
 B. Program 1 displays false and Program 2 displays true
 C. Program 1 displays true and Program 2 displays false
 D. Program 1 displays false and Program 2 displays false

17 What modifier should you use on the members of a class so that they are not accessible to another class in a different package, but are accessible to any subclasses in any package? [1 Mark]

- A. public
 B. private
 C. protected
 D. Use the default modifier.

18 Which of the following statements is false? [1 Mark]

- A. A public class can be accessed by a class from a different package.
 B. A private method cannot be accessed by a class in a different package.
 C. A protected method can be accessed by a subclass in a different package.
 D. A method with no visibility modifier can be accessed by a class in a different package.

19 Which component cannot be added to a container? [1 Mark]

- A. JPanel
 B. JButton
 C. JFrame
 D. JComponent

20 How many frames are displayed? [2 Marks]

```

import javax.swing.*;
public class Test {
public static void main(String[] args) {
JFrame f1 = new JFrame("My Frame");
JFrame f2 = f1;
JFrame f3 = f2;
f1.setVisible(true);
f2.setVisible(true);
f3.setVisible(true); }
}

```

- A. 1.
 B. 2.
 C. 3.
 D. 0.

21 What layout manager should you use so that every component occupies the same size in the container? [1 Mark]

- A. a FlowLayout
 B. a GridLayout
 C. a BorderLayout
 D. any layout

22 Suppose a JFrame uses the GridLayout(2, 2). If you add six buttons to the frame, how many columns are displayed? [1 Mark]

- A. 1
 B. 2
 C. 3
 D. 4

23 To set a FlowLayout in panel jp, you can use the method _____. (Choose Two) [1 Mark]

- A. jp.setLayout(new FlowLayout());
 B. jp.setLayout(new FlowLayout(FlowLayout.CENTER));
 C. jp.setLayout(new FlowLayout(FlowLayout.CENTER));
 D. jp.setLayout(FlowLayout());

24 The default layout out of a contentPane in a JFrame is _____. [1 Mark]

- A. FlowLayout
 B. GridLayout
 C. BorderLayout
 D. None

25 To create a JPanel of the BorderLayout, use _____. [1 Mark]

- A. JPanel p = new JPanel()
 B. JPanel p = new JPanel(BorderLayout());
 C. JPanel p = new JPanel(new BorderLayout());
 D. JPanel p = new JPanel().setLayout(new BorderLayout());

26 Show the output of the following code? [2 Marks]

```

import javax.swing.*;

```

```

public class Test {
    public static void main(String[] args) {
        JButton jbtOK = new JButton("OK");
        System.out.print(jbtOK.isVisible() + " , ");
        JFrame frame = new JFrame();
        System.out.println(frame.isVisible()); }
    }

```

- A. true, true B. true, false C. false, true D. false, false

27 The header for the paintComponent method is _____.[1 Mark]

- A. private void paintComponent(Graphics g) B. protected void paintComponent(Graphics g)
 C. public void paintComponent(Graphics g) D. protected void paintComponent()

28 You should override the _____ method to draw things on a Swing component. [1 Mark]

- A. repaint() B. update() C. paintComponent() D. init()

29 To draw graphics, it is better to define a class that extends _____ and override the paintComponent method. [1 Mark]

- A. JLabel B. JButton C. JPanel D. JComponent

30 Analyze the following code. [2 Marks]

```

import java.awt.*;
import javax.swing.*;
public class Test extends JFrame {
    public Test() {
        add(new MyDrawing("Welcome to Java!")); }
    public static void main(String[] args) {
        JFrame frame = new JFrame();
        frame.setSize(300, 300);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setVisible(true); }
    }
class MyDrawing extends JPanel {
    String message;
    public MyDrawing(String message) {
        this.message = message; }
    public void paintComponent(Graphics g) {
        super.paintComponent(g);
        g.drawString(message, 20 ,20); }
    }

```

- A. The program runs fine and displays Welcome to Java!
 B. The program would display Welcome to Java! if new JFrame() is replaced by Test().
 C. The program would display Welcome to Java! if new JFrame() is replaced by new Test().
 D. The program would display Welcome to Java! if new JFrame() is replaced by new Test("My Frame").

31 The interface _____ should be implemented to listen for a button action event. [1 Mark]

- A. MouseListener B. ActionListener C. FocusListener D. WindowListener E. ContainerListener

32 Suppose A is an anonymous inner class in Test. A is compiled into a file named _____.[1 Mark]

- A. A.class B. Test\$A.class C. A\$Test.class D. Test\$1.class E. Test&1.class

33 Analyze the following code. [2 Marks]

```

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
public class Test extends JFrame {
    public void Test() {

```

```

JButton jbtOK = new JButton("OK");
add(jbtOK);
jbtOK.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        System.out.println("The OK button is clicked");
    } });
}
public static void main(String[] args) {
    JFrame frame = new Test();
    frame.setSize(300, 300);
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.setVisible(true); }
}

```

- A. When you run the program, the button is not displayed, because the constructor is declared wrong. It should be declared public Test(), not public void Test().
- B. The program has a compile error because no listeners are registered with jbtOK.
- C. The program has a runtime error because no listeners are registered with jbtOK.
- D. The message "The OK button is clicked" is displayed when you click the OK button.

34 Analyze the following code: [3 Marks]

```

import javax.swing.*;
import java.awt.*;
public class Test extends JFrame {
    public Test() {
        ImageIcon uslcon = new ImageIcon("image/uslcon.gif");
        JButton jbt1 = new JButton(yslcon);
        JButton jbt2 = new JButton(yslcon);
        JPanel p1 = new JPanel();
        p1.add(jbt1);
        JPanel p2 = new JPanel();
        p2.add(jbt2);
        JPanel p3 = new JPanel();
        p2.add(jbt1);
        add(p1, BorderLayout.NORTH);
        add(p2, BorderLayout.SOUTH);
        add(p3, BorderLayout.CENTER);
    }
    public static void main(String[] args) {
        JFrame frame = new Test();
        frame.setSize(200, 100);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setVisible(true); }
}

```

- A. Two buttons displayed with different icons.
- B. Three buttons displayed with the same icon.
- C. Only jbt1 displays the icon and jbt2 does not display the icon.
- D. Two buttons displayed with the same icon.

Answer Paper

A large rectangular area containing numerous horizontal dotted lines, intended for writing answers.

Answer Paper

A large rectangular area containing numerous horizontal dotted lines, intended for writing answers.

Answer the following questions:

Section I

[Total 48]

Q (1): Multiple choice questions Place the answer on the **special MCQ form**.

Section I : Key answer for the multiple choice questions							
Questions	1	2	3	4	5	6	7
Answer Key	BD	B	C	C	B	C	B
Questions	8	9	10	11	12	13	14
Answer Key	B	BC	B	D	AC	A	D
Questions	15	16	17	18	19	20	21
Answer Key	A	B	C	D	C	A	B
Questions	22	23	24	25	26	27	28
Answer Key	C	AB	C	C	B	B	C
Questions	29	30	31	32	33	34	
Answer Key	C	C	B	D	A	D	

1- what is wrong with the following program? [4 Marks]

```

1 public class ShowErrors {
2     public static void main(String[] args) {
3         ShowErrors t = new ShowErrors(5);
4     }
5 }

```

(a)

```

1 public class ShowErrors {
2     public static void main(String[] args) {
3         ShowErrors t = new ShowErrors();
4         t.x();
5     }
6 }

```

(b)

```

1 public class ShowErrors {
2     public void method1() {
3         Circle c;
4         System.out.println("What is radius ")
5             + c.getRadius();
6         c = new Circle();
7     }
8 }

```

(c)

```

1 public class ShowErrors {
2     public static void main(String[] args) {
3         C c = new C(5.0);
4         System.out.println(c.value);
5     }
6 }
7
8 class C {
9     int value = 2;
10 }

```

(d)

2- What is wrong in the following code? [2 Marks]

```

1 class Test {
2     public static void main(String[] args) {
3         A a = new A();
4         a.print();
5     }
6 }
7
8 class A {
9     String s;
10
11     A(String s) {
12         this.s = s;
13     }
14
15     public void print() {
16         System.out.print(s);
17     }
18 }

```

3- Show the output of following program: [3 Marks]

```

1 public class Test {
2     public static void main(String[] args) {
3         A a = new A(3);
4     }
5 }
6
7 class A extends B {
8     public A(int t) {
9         System.out.println("A's constructor is invoked");
10    }
11 }
12
13 class B {
14     public B() {
15         System.out.println("B's constructor is invoked");
16     }
17 }

```

Is the no-arg constructor of Object invoked when new A(3) is invoked?..... **Answer Paper****Ans Q1:**

(a) On Line 3, new ShowErrors(5) attempts to create an instance using a constructor ShowErrors(int), but the ShowErrors class does not have such a constructor. That is an error.

(b) On Line 4, t.x() is invoked, but the ShowErrors class does not have the method named x(). That is an error.

(c) The program compiles fine, but it has a runtime error because variable c is null when the println statement is executed.

(d) new C(5.0) does not match any constructors in class C. The program has a compilation error because class C does not have a constructor with a double argument.

Ans Q2:

The program does not compile because new A() is used in class Test, but class A does not have a default constructor.

Ans Q3:

B's constructor is invoked

A's constructor is invoked

The default constructor of Object is invoked, when new A(3) is invoked. The Object's constructor is invoked before any statements in B's constructor are executed.

4- How do you set background color, foreground color, font, and tool tip text on a Swing GUI component? Why is the tool tip text not displayed in the following code? [3 Marks]

```
1 import javax.swing.*;
2
3 public class Test extends JFrame {
4     private JButton jbtOK = new JButton("OK");
5
6     public static void main(String[] args) {
7         // Create a frame and set its properties
8         JFrame frame = new Test();
9         frame.setTitle("Logic Error");
10        frame.setSize(200, 100);
11        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
12        frame.setVisible(true);
13    }
14
15    public Test() {
16        jbtOK.setToolTipText("This is a button");
17        add(new JButton("OK"));
18    } }
```

5- Show the output of the following code: [3 Marks]

```
import javax.swing.*;
public class Test {
    public static void main(String[] args) {
        JButton jbtOK = new JButton("OK");
        System.out.println(jbtOK.isVisible());
        JFrame frame = new JFrame();
        System.out.println(frame.isVisible());
    } }
```

6- Show the output of the following code: [3 Marks]

```
1 public class Faculty extends Employee{
2     public static void main(String[] args) {
3         new Faculty();
4     }
5
6     public Faculty() {
7         System.out.println("(4) Performs Faculty's tasks");
8     }
9 }
11 class Employee extends Person {
12     public Employee() {
13         this("(2) Invoke Employee's overloaded constructor");
14         System.out.println("(3) Performs Employee's tasks ");
15     }
16
17     public Employee(String s) {
18         System.out.println(s);
19     }
20 }
22 class Person {
23     public Person() {
24         System.out.println("(1) Performs Person's tasks");
25     }
26 }
```

Answer Paper

Ans Q4:

The tool tip text is not displayed, because Line 16 sets a tool tip text on jbtOK, but Line 17 adds a new button (different from jbtOK) to the content pane of the frame.

Ans Q5:

True
false

Ans Q6:

- (1) Performs Person's tasks
- (2) Invoke Employee's overloaded constructor
- (3) Performs Employee's tasks
- (4) Performs Faculty's tasks

Section III you should attempt 2 out of section III (questions 2, 3, 4 and 5) [Total 24]

Q(2): (Checking Palindromes and Counts Occurrences), a string is a palindrome if it reads the same forward and backward. The words “mom,” “dad,” and “noon,” for instance, are all palindromes. **Outline a program that prompts the user to enter a string and reports whether the string is a palindrome and counts the number of occurrences of each letter in the string regardless of case.** Hint: one solution is to check whether the first character in the string is the same as the last character. If so, check whether the second character is the same as the second-to-last character. This process continues until a mismatch is found or all the characters in the string are checked, except for the middle character if the string has an odd number of characters. **[12 Marks]**

Ans Q2:

```
import java.util.Scanner;
public class CheckPalindrome {
    public static void main(String[] args) {
        // Create a Scanner
        Scanner input = new Scanner(System.in);
        // Prompt the user to enter a string
        System.out.print("Enter a string: ");
        String s = input.nextLine();
        if (isPalindrome(s)) {
            System.out.println(s + " is a palindrome");
        } else {
            System.out.println(s + " is not a palindrome");
        }
    }
    // Display results
    int[] counts = countLetters(s.toLowerCase());
    for (int i = 0; i < counts.length; i++) {
        if (counts[i] != 0) {
            System.out.println((char) ('a' + i) + " appears "
                + counts[i] + ((counts[i] == 1) ? " time" : " times"));
        }
    }
}
public static boolean isPalindrome(String s) {
    // The index of the first character in the string
    int low = 0;
    // The index of the last character in the string
    int high = s.length() - 1;
    while (low < high) {
        if (s.charAt(low) != s.charAt(high)) {
            return false; // Not a palindrome
        }
        low++;
        high--;
    }
    return true; // The string is a palindrome
}
public static int[] countLetters(String s) {
    int[] counts = new int[26];
    for (int i = 0; i < s.length(); i++) {
        if (Character.isLetter(s.charAt(i))) {
            counts[s.charAt(i) - 'a']++;
        }
    }
    return counts;
}
}
```

4 marks

3 marks

5 Marks

4 Marks

Q(3): Outline a program demonstrates Swing common features. The example creates a panel **p1** to hold three buttons and a panel **p2** to hold two labels. The background of the button **jbtLeft** is set to white and the foreground of the button **jbtCenter** is set to green. The tool tip of the button **jbtRight** is set (This is the Right Button). Titled borders are set on panels **p1** and **p2** and line borders are set on the labels, as shown in Fig. (1). **[12 Marks]**

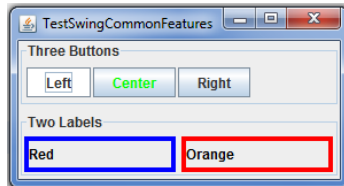


Figure 1

Ans Q3:

```

1 import java.awt.*;
2 import javax.swing.*;
3 import javax.swing.border.*;
4
5 public class TestSwingCommonFeatures extends JFrame {
6     public TestSwingCommonFeatures() {
7         // Create a panel to group three buttons
8         JPanel p1 = new JPanel(new FlowLayout(FlowLayout.LEFT, 2, 2));
9         JButton jbtLeft = new JButton("Left");
10        JButton jbtCenter = new JButton("Center");
11        JButton jbtRight = new JButton("Right");
12        jbtLeft.setBackground(Color.WHITE);
13        jbtCenter.setForeground(Color.GREEN);
14        jbtRight.setToolTipText("This is the Right button");
15        p1.add(jbtLeft);
16        p1.add(jbtCenter);
17        p1.add(jbtRight);
18        p1.setBorder(new TitledBorder("Three Buttons"));
19
20        // Create a font and a line border
21        Font largeFont = new Font("TimesRoman", Font.BOLD, 20);
22        Border lineBorder = new LineBorder(Color.BLACK, 2);
23
24        // Create a panel to group two labels
25        JPanel p2 = new JPanel(new GridLayout(1, 2, 5, 5));
26        JLabel jlblRed = new JLabel("Red");
27        JLabel jlblOrange = new JLabel("Orange");
28        jlblRed.setForeground(Color.RED);
29        jlblOrange.setForeground(Color.ORANGE);
30        jlblRed.setFont(largeFont);
31        jlblOrange.setFont(largeFont);
32        jlblRed.setBorder(lineBorder);
33        jlblOrange.setBorder(lineBorder);
34        p2.add(jlblRed);
35        p2.add(jlblOrange);
36        p2.setBorder(new TitledBorder("Two Labels"));
37
38        // Add two panels to the frame
39        setLayout(new GridLayout(2, 1, 5, 5));
40        add(p1);
41        add(p2);
42    }
43
44    public static void main(String[] args) {
45        // Create a frame and set its properties
46        JFrame frame = new TestSwingCommonFeatures();
47        frame.setTitle("TestSwingCommonFeatures");
48        frame.setSize(300, 150);
49        frame.setLocationRelativeTo(null); // Center the frame
50        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
51        frame.setVisible(true);
52    }
53 }

```

1 mark

4 Marks

4 Marks

1 mark

2 Marks

Q(4): (Creating four fans) Outline a program that places four fans in a frame of GridLayout with two rows and two columns, as shown in Fig. (2). **[12 Marks]**

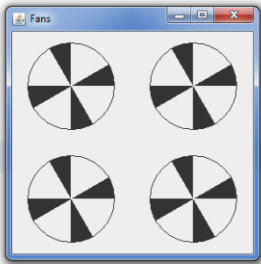


Figure 2

Ans Q4:

```

import javax.swing.*;
import java.awt.*;
// The class for drawing arcs on a panel
public class Fans extends JFrame {
    public Fans () {
        setTitle("Fans");
        setLayout(new GridLayout(2, 2, 5, 5));
        add(new ArcsPanel());
        add(new ArcsPanel());
        add(new ArcsPanel());
        add(new ArcsPanel());
    }
    /** Main method */
    public static void main(String[] args) {
        Fans frame = new Fans ();
        frame.setSize(300, 300);
        frame.setLocationRelativeTo(null); // Center the frame
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setVisible(true);
    }
}
class ArcsPanel extends JPanel {
    // Draw four blades of a fan
    protected void paintComponent(Graphics g) {
        super.paintComponent(g);
        int xCenter = getWidth() / 2;
        int yCenter = getHeight() / 2;
        int radius = (int) (Math.min(getWidth(), getHeight()) * 0.4);
        int x = xCenter - radius;
        int y = yCenter - radius;
        //setBackground(Color.WHITE);
        g.fillArc(x, y, 2 * radius, 2 * radius, 0, 30);
        g.fillArc(x, y, 2 * radius, 2 * radius, 90, 30);
        g.fillArc(x, y, 2 * radius, 2 * radius, 180, 30);
        g.fillArc(x, y, 2 * radius, 2 * radius, 270, 30);

        g.drawOval ((int)(x), (int)(y),(int)(2 * radius), (int)(2 * radius));
    }
}

```

1 mark

4 Marks

3 Marks

4 Marks

(5): (Moving the Rectangle) **Outline a program** that moves the rectangle in a panel. You should define a panel class for displaying the rectangle and provide the methods for moving the button left, right, up, and down, as shown in Fig. (3). **[12 Marks]**

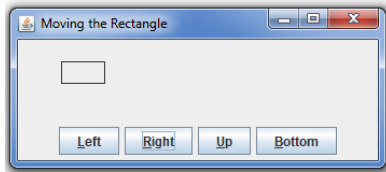


Figure 3

Ans Q5:

```
import java.awt.*;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import javax.swing.*;
public class MoveRect extends JFrame {
    protected RectPanel rect = new RectPanel();
    private JButton jbtLeft = new JButton("Left");
    private JButton jbtRight = new JButton("Right");
    private JButton jbtUp = new JButton("Up");
    private JButton jbtBottom = new JButton("Bottom");
    public static void main(String[] args) {
        MoveRect frame = new MoveRect();
        frame.setTitle("Moving the Rectangle");
        frame.setSize(350, 150);
        frame.setLocationRelativeTo(null); // Center the frame
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setVisible(true);
    }
    public MoveRect() {
        JPanel jpButtons = new JPanel();
        jpButtons.add(jbtLeft);
        jpButtons.add(jbtRight);
        jpButtons.add(jbtUp);
        jpButtons.add(jbtBottom);
        jbtLeft.setMnemonic('L');
        jbtRight.setMnemonic('R');
        jbtUp.setMnemonic('U');
        jbtBottom.setMnemonic('B');
        jbtLeft.setToolTipText("Move message to left");
        jbtRight.setToolTipText("Move message to right");
        setLayout(new BorderLayout());
        add(rect, BorderLayout.CENTER);
        add(jpButtons, BorderLayout.SOUTH);
        jbtLeft.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                rect.moveLeft();
            }
        });
        jbtRight.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                rect.moveRight();
            }
        });
        jbtUp.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                rect.moveUp();
            }
        });
        jbtBottom.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                rect.moveDown();
            }
        });
    }
}
```

1 mark

4 Marks

4 Marks

```
class RectPanel extends JPanel { // Inner class
    private int w = 5;
    private int h = 5;
    private int xCoordinate = getWidth() / 2;
    private int yCoordinate = getHeight() / 2;
    private int interval = 10;
    public void moveLeft() {
        xCoordinate -= interval;
        repaint();
    }
    public void moveRight() {
        xCoordinate += interval;
        repaint();
    }
    public void moveUp() {
        yCoordinate -= interval;
        repaint();
    }
    public void moveDown() {
        yCoordinate += interval;
        repaint();
    }
    protected void paintComponent(Graphics g) {
        super.paintComponent(g);
        g.drawRect(xCoordinate, yCoordinate,
            8 * w, 4 * h);
    }
}
```

3 Marks

..... Good Luck