(d) Create two Car objects and add them to a Vehicle array with the name of VehicleArray. Use a loop to iterate through the array to show the finally comparable results.

(5 marks)

(e) Illustrate how the concepts of dynamic binding works by invoking the toString() method or the equals() method.

(5 marks)

(2) Create a class called ShowWindow which is the subclass of the JFrame class.

(import java.awt.*; import javax.swing.*; import java.awt.event.*;)

(a) Add a JPanel to ShowWindow .

(4 marks)

(b) Add two JButton objects called "Sart" and "Finish" to the panel. (4 marks)

(c) Set the title, name and size of the Show Window. (3 marks)

(d)Provide an implementation to indicate which button is pressed.

(5 marks)

The ActionListener interface has one method:

```
public void actionPerformed(ActionEvent e){}
```

(e) Use this example to explain the principles of event-driven programming.

(4 marks)

```
public class Test
{
public static void main(String args[])
    {
    M m1 = new M();
    M m2 = new M();
  System.out.print(m1.equals(m2));
  System.out.print(m1.toString());
    }
}
```

(2)
```
public class A extends B
  {
        A()
        {
        System.out.println(a);
      }
     int a=0;
  }
    public class B
    {
      B(String b)
       {
       System.out.println(b);
      }
      B(int b)
      {
       System.out.println(b);
    }
      int b=1;
  }
  public class Test
    {
        public static void main(String args[])
          {
        A a= new A();
    }
}
```

Is there any error when the above code is implemented? If not, please give the output. If you feel there is any mistake in it, please give the right codes in the suitable location and then show the output.

(3)
```
public class Test {
```

```
    public static void main(String[] args)
{
try {
            System.out.println("Welcome to Java");
            int i= 0;
            int y = 10/i;
            System.out.println("Welcome to Earth");
        }
    catch (RuntimeException e)
    {
       System.out.println("Wrong Calculator");
    }
finally {
            System.out.println("End of the block");
            }
    }
}
```

4. Writing Codes (45marks)

(1) Design a class named Car that has attributes named as Color(type of String), Place of production (type of String), and Price(type of double). The supper class the Car is named as Vehicle and the subclass of Car is named as Truck. Suppose two vehicles be comparable by their price and thus the Vehicle class should implement the Comparable interface. The Comparable interface has one method: public int compareTo(Object o). The Vehicle class should also implement the Cloneable interface. Therefore it should implement the method: public Object clone().

(a) Draw a UML class diagram to illustrate the design of the above class and the relationship with the two interfaces. (5 marks)

(b) The Car class should have two types of constructors and three methods getColor(), getPlace() and getPrice(), which should be abstract in the class Vehicle. (6 marks)

(c) Override the toString() and equals() methods in the class Truck.

```
public String toString(){}
public boolean equals(Object obj)
{
        return (this == obj);
}
```

(4 marks)

Concept Explanation (10 marks)

) Class abstraction and Encapsulation (2 marks)

) Exception and Error (2 marks)

) Polymorphism (2 marks)

4) Casting: Implicit casting and Explicit casting (3 marks)

5) Event (1 marks)

2. Answer Questions (30 marks)

(1) Please list at least four features of the constructor (4 marks)

(2) Please list the names and symbolic representations of the relationships between classes that you have learned, and briefly describe them. (4marks)

(3) How can you judge a class to be immutable? (3 marks)

(4) Can an abstract method be contained in a non-abstract class? Suppose A is the superclass and B is its subclass, if B is abstract, is A necessarily to be abstract? If A is abstract, is B necessarily to be abstract? If not, how can you make sure B is non-abstract. (6marks)

(5) Please compare the usage of the key words "this" and "super". (4 marks)

(6) Please give at least five types of visibility modifiers you have learned, and explain clearly the accessibility of them. (5 marks)

(7) Please explain the differences between the dynamic binding and the method matching. (4 marks)

3. Reading Codes and Show the Outputs (15 marks)

1) class M

```
{
  int p;
  int q;
M()
{
    p = 1;
    q = 2;
  }
}
```