**Theory Task**

1 - Find 3 examples of passing an object reference to a method.

|  |
| --- |
| // Java program to demonstrate objects  // passing to methods.  class ObjectPassDemo  {  int a, b;    ObjectPassDemo(int i, int j)  {  a = i;  b = j;  }    // return true if o is equal to the invoking  // object notice an object is passed as an  // argument to method  boolean equalTo(ObjectPassDemo o)  {  return (o.a == a && o.b == b);  }  }    // Driver class  public class Test  {  public static void main(String args[])  {  ObjectPassDemo ob1 = new ObjectPassDemo(100, 22);  ObjectPassDemo ob2 = new ObjectPassDemo(100, 22);  ObjectPassDemo ob3 = new ObjectPassDemo(-1, -1);    System.out.println("ob1 == ob2: " + ob1.equalTo(ob2));  System.out.println("ob1 == ob3: " + ob1.equalTo(ob3));  }  } |

Output:

ob1 == ob2: true

ob1 == ob3: false

|  |
| --- |
| // Java program to demonstrate one object to  // initialize another  class Box  {  double width, height, depth;    // Notice this constructor. It takes an  // object of type Box. This constructor use  // one object to initialize another  Box(Box ob)  {  width = ob.width;  height = ob.height;  depth = ob.depth;  }    // constructor used when all dimensions  // specified  Box(double w, double h, double d)  {  width = w;  height = h;  depth = d;  }    // compute and return volume  double volume()  {  return width \* height \* depth;  }  }    // driver class  public class Test  {  public static void main(String args[])  {  // creating a box with all dimensions specified  Box mybox = new Box(10, 20, 15);    // creating a copy of mybox  Box myclone = new Box(mybox);    double vol;    // get volume of mybox  vol = mybox.volume();  System.out.println("Volume of mybox is " + vol);    // get volume of myclone  vol = myclone.volume();  System.out.println("Volume of myclone is " + vol);  }  } |

Output:

Volume of mybox is 3000.0

Volume of myclone is 3000.0

|  |
| --- |
| // Java program to demonstrate returning  // of objects  class ObjectReturnDemo  {  int a;    ObjectReturnDemo(int i)  {  a = i;  }    // This method returns an object  ObjectReturnDemo incrByTen()  {  ObjectReturnDemo temp =  new ObjectReturnDemo(a+10);  return temp;  }  }    // Driver class  public class Test  {  public static void main(String args[])  {  ObjectReturnDemo ob1 = new ObjectReturnDemo(2);  ObjectReturnDemo ob2;    ob2 = ob1.incrByTen();    System.out.println("ob1.a: " + ob1.a);  System.out.println("ob2.a: " + ob2.a);  }  } |

Output:

ob1.a: 2

ob2.a: 12

2 - Describe how message passing between subject and object is done.

Message Passing in terms of computers is communication between processes. It is a form of communication used in object-oriented programming as well as parallel programming. Message passing in Java is like sending an object i.e. message from one thread to another thread. It is used when threads do not have shared memory and are unable to share monitors or semaphores or any other shared variables to communicate.

3 - Which type of relationship exists between the subject and the objects, association, aggregation or composition? Read this resource to find out

<https://www.visual-paradigm.com/guide/uml-unified-modeling-language/uml-aggregation-vs-composition/>.

4 - Draw out the UML class diagram for the program.

