**LSET Preparation – Java**

**Mock Test – V (Apr 10, 2014)**

**Duration: 60 Minutes**

**[Total Marks: 70]**

Name : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Please fill up the answers in the table below.

**Answers - Java:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Question No** | **Answer** | **Question No** | **Answer** |
| 1 | 434 | 14 | 7 7 |
| 2 | b h hn x | 15 | 42 |
| 3 | b | 16 | 2814 |
| 4 | b | 17 | DOG Sounds = woof  CAT Sounds = meooww  FISH Sounds = burble |
| 5 | b | 18 | 4 4 |
| 6 | a | 19 | The value of passing is: 3  The value of passing object is: default  The value of passing object is: modified |
| 7 | a | 20 | a |
| 8 | d | 21 | d |
| 9 | a | 22 | d |
| 10 | b | 23 | d |
| 11 | 9 49 86 foo | 24 | Return Value : Tutorialspoint.com |
| 12 | B | 25 | Inside move::Subclass  Inside move::Super  move value= 0 |
| 13 | DOG Sounds = woof  CAT Sounds = meow  FISH Sounds = burble |  | |

**Section I: Java Marks: 70**

|  |  |  |
| --- | --- | --- |
| Q1 | class A {  }  class B extends A {  }  public class ComingThru {  static String s = "-";  public static void main(String[] args) {  A[] aa = new A[2];  B[] ba = new B[2];  sifter(aa);  sifter(ba);  sifter(7);  System.out.println(s);  }  static void sifter(A[]... a2) {  s += "1";  }  static void sifter(B[]... b1) {  s += "2";  }  static void sifter(B[] b1) {  s += "3";  }  static void sifter(Object o) {  s += "4";  }  }  What is the output of above program? | 2 |
| Q2 | class Building {  Building() {  System.out.print("b ");  }  Building(String name) {  this();  System.out.print("bn " + name);  }  }  public class House extends Building {  House() {  System.out.print("h ");  }    House(String name) {  this();  System.out.print("hn " + name);  }  public static void main(String[] args) {  new House("x ");  }  }  What is the output of above program? | 4 |
| Q3 | import java.util.ArrayList;  import java.util.List;  public class GenericTest1 {  public static void main(String...args){  List<Rat> ratlist = new ArrayList<Rat>();  ratlist.add(new Rat());  ratlist.add(new Rat());  ratlist.add(new Rat());  doNothing(ratlist);  }    static void doNothing(List<Animalss> animal){  System.out.println("Inside doNothing");  }      }  class Animalss{    }  class Rat extends Animalss{    }  class Cat extends Animalss{    }  What is the output of above program?   1. Inside doNothing 2. Compilation fails 3. Runtime Exception is thrown 4. Multiple compilation errors | 4 |
| Q4 | import java.util.ArrayList;  import java.util.List;  public class GenericTest1 {  public static void main(String...args){  List<Rat> ratlist = new ArrayList<Rat>();  ratlist.add(new Rat());  ratlist.add(new Rat());  ratlist.add(new Rat());  doNothing(ratlist);  }    static void doNothing(List<? extends Animalss> animal){  System.out.println("Inside doNothing");  }      }  class Animalss{    }  class Rat extends Animalss{    }  class Cat extends Animalss{    }    What is the output of above program?   1. Inside doNothing 2. Compilation fails 3. Runtime Exception is thrown 4. Multiple compilation errors | 4 |
| Q5 | import java.util.ArrayList;  import java.util.List;  public class GenericTest1 {  public static void main(String...args){  List<Rat> ratlist = new ArrayList<Rat>();  ratlist.add(new Rat());  ratlist.add(new Rat());  ratlist.add(new Rat());  doNothing(ratlist);//Line 10  }    static void doNothing(List<? extends Animalss> animal){  System.out.println("Inside doNothing");  animal.add(new Rat());//Line 15  }      }  class Animalss{    }  class Rat extends Animalss{    }  class Cat extends Animalss{    }  What is the output of above program?   1. Inside doNothing 2. Compilation fails at line 15 3. Runtime Exception is thrown 4. Compilation fails at line 10 | 4 |
| Q6 | import java.util.ArrayList;  import java.util.List;  public class GenericTest2 {  public static void main(String ...strings){  List<WildAnimal> lionlist = new ArrayList<WildAnimal>();  lionlist.add(new Lion());  lionlist.add(new Lion());  lionlist.add(new Lion());  doNothing(lionlist);//Line 10  }    static void doNothing(List<? super WildAnimal> wildAnimal){  System.out.println("Inside doNothing WildAnimal");  }  }  class WildAnimal{    }  class Lion extends WildAnimal{    }  class Tiger extends WildAnimal{    }  What is the output of above program?   1. Inside doNothing WildAnimal 2. Multiple Compilation errors 3. Runtime Exception is thrown 4. Compilation fails at line 10 | 4 |
| Q7 | import java.util.ArrayList;  import java.util.List;  public class GenericTest2 {  public static void main(String ...strings){  List<WildAnimal> lionlist = new ArrayList<WildAnimal>();  lionlist.add(new Lion());  lionlist.add(new Lion());  lionlist.add(new Lion());  doNothing(lionlist);  }    static void doNothing(List<? super WildAnimal> wildAnimal){  System.out.println("Inside doNothing WildAnimal");  wildAnimal.add(new Tiger());  }  }  class WildAnimal{    }  class Lion extends WildAnimal{    }  class Tiger extends WildAnimal{    }  What will be output of above program?   1. Inside doNothing WildAnimal 2. Multiple Compilation errors 3. Runtime Exception is thrown 4. Compilation fails at line 10 | 4 |
| Q8 | import java.util.ArrayList;  import java.util.List;  public class GenericTest2 {  public static void main(String ...strings){  List<Lion> lionlist = new ArrayList<Lion>();  lionlist.add(new Lion());  lionlist.add(new Lion());  lionlist.add(new Lion());  doNothing(lionlist);//Line 10  }    static void doNothing(List<? super WildAnimal> wildAnimal){  System.out.println("Inside doNothing WildAnimal");  wildAnimal.add(new Tiger());//Line 15  }  }  class WildAnimal{    }  class Lion extends WildAnimal{    }  class Tiger extends WildAnimal{    }  What will be output of above program?   1. Inside doNothing 2. Compilation fails at line 15 3. Runtime Exception is thrown 4. Compilation fails at line 10 | 4 |
| Q9 | import java.util.ArrayList;  import java.util.List;  public class GenericTest2 {  public static void main(String ...strings){  List<Object> lionlist = new ArrayList<Object>();  lionlist.add(new Lion());  lionlist.add(new Lion());  lionlist.add(new Lion());  doNothing(lionlist);  }      static void doNothing(List<?> object){  System.out.println("Inside doNothing WildAnimal");  }  }  class WildAnimal{    }  class Lion extends WildAnimal{    }  class Tiger extends WildAnimal{    }  What will be output of above program?   1. Inside doNothing WildAnimal 2. Compilation fails 3. Runtime Exception is thrown 4. Multiple Compilation Errors | 2 |
| Q10 | import java.util.ArrayList;  import java.util.List;  public class GenericTest2 {  public static void main(String ...strings){  List<Object> lionlist = new ArrayList<Object>();  lionlist.add(new Lion());  lionlist.add(new Lion());  lionlist.add(new Lion());  doNothing(lionlist);  }      static void doNothing(List<?> object){  System.out.println("Inside doNothing WildAnimal");  object.add(new Lion());//Line 17  }  }  class WildAnimal{    }  class Lion extends WildAnimal{    }  class Tiger extends WildAnimal{    }  What will be output of above program?   1. Inside doNothing WildAnimal 2. Compilation fails at Line 17 3. Runtime Exception is thrown 4. Multiple Compilation Errors | 2 |
| Q11 | class Feline {  public static void main(String[] args) {  Long x = 42L;  Long y = 44L;  System.out.print(7 + 2 + " ");  System.out.print(y + 5 + " ");  System.out.println(x + y + " "+foo());  }  static String foo() {  return "foo";  }  }  What will be output of above program? | 4 |
| Q12 | interface Hungry<E> {  void munch(E x);  }  interface Carnivore<E extends Animal> extends Hungry<E> {  }  interface Herbivore<E extends Plant> extends Hungry<E> {  }  abstract class Plant {  }  class Grass extends Plant {  }  abstract class Animal {  }  class Sheep extends Animal implements Herbivore<Sheep> {  public void munch(Sheep x) {  }  }  class Wolf extends Animal implements Carnivore<Sheep> {  public void munch(Sheep x) {  }  }  Which of the following changes (taken separately) would allow this code to compile?  (Choose all that apply.)  A. Change the Carnivore interface to  interface Carnivore<E extends Plant> extends Hungry<E> {}  B. Change the Herbivore interface to  interface Herbivore<E extends Animal> extends Hungry<E> {}  C. Change the Sheep class to  class Sheep extends Animal implements Herbivore<Plant> {  public void munch(Grass x) {}  }  D. Change the Sheep class to  class Sheep extends Plant implements Carnivore<Wolf> {  public void munch(Wolf x) {}  }  E. Change the Wolf class to  class Wolf extends Animal implements Herbivore<Grass> {  public void munch(Grass x) {}  }  F. No changes are necessary | 4 |
| Q13 | public class EnumTest1 {  static Animals animals;  public static void main(String... args) {  for (Animals animal : Animals.values()) {  System.out.println(animal + " " + "Sounds = " +  animal.sounds);  }  }  }  enum Animals {  DOG("woof"), CAT("meow"), FISH("burble");  String sounds;  Animals(String sounds) {  this.sounds = sounds;  }  }  What is output of above program? | 4 |
| Q14 | class Birds {  int i = 14;  }  class Raptor extends Birds {  Raptor(){  i = 28;  }  }  class Hawk extends Raptor {  Hawk(){  i = 7;  }    public static void main(String[] args) {  System.out.print(((Raptor)new Hawk()).i+" ");  System.out.print(((Birds)new Hawk()).i);  }  }  What will be output of above program? | 2 |
| Q15 | class Bird {  int i = 14;  }  class Raptor extends Bird {  int i = 28;  }  class Hawk extends Raptor {  int i = 7;  public static void main(String[] args) {  System.out.print(((Raptor)new Hawk()).i+((Bird)new Hawk()).i);  }  }  What will be output of above program? | 2 |
| Q16 | class Bird {  int i = 14;  }  class Raptor extends Bird {  int i = 28;  }  class Hawk extends Raptor {  int i = 7;  public static void main(String[] args) {  System.out.print(""+((Raptor)new Hawk()).i+((Bird)new Hawk()).i);  }  }  What will be output of above program? | 2 |
| Q17 | public class EnumTest1 {  static Animals animals;  public static void main(String... args) {  for (Animals animal : Animals.values()) {  System.out.println(animal.name() + " " + "Sounds = " +  animal.sounds);  }  }  }  enum Animals {  DOG("woof"), CAT("meooww"), FISH("burble");  String sounds;  Animals(String sounds) {  this.sounds = sounds;  }  }  What will be output of above program? | 2 |
| Q18 | class Box {  int size;  Box(int s) {  size = s;  }  }  class Laser {  public static void main(String[] args) {  Box b1 = new Box(5);  Box[] ba = go(b1, new Box(6));  ba[0] = b1;  for (Box b : ba)  System.out.print(b.size + " ");  }  static Box[] go(Box b1, Box b2) {  b1.size = 4;  Box[] ma = { b2, b1 };  return ma;  }  }  What will be output of above program? | 2 |
| Q19 | public class PassByValueTest {  String s = "default";  public static void main(String[] args) {  int passing = 3;  Receiving(passing);  System.out.println("The value of passing is: " + passing);  PassByValueTest obj = new PassByValueTest();  Receiving(obj);  System.out.println("The value of passing object is: " + (obj.s));  receiving(obj);  System.out.println("The value of passing object is: " +  (obj.s));  }  public static void Receiving(int var) {  var = var + 2;  }  public static void Receiving(Object obj){  obj = new PassByValueTest();  ((PassByValueTest)obj).s = "modified";  }    public static void receiving(PassByValueTest obj){  obj.s = "modified";  }  }  What will be output of above program? | 2 |
| Q20 | public class ExceptionTest {  public static void main(String...strings ){  Supertesting superobj = new Subclasstest();  superobj.method();  }  }  class Supertesting{  public void method() throws NullPointerException{  System.out.println("Inside method::Supertesting");  }  }  class Subclasstest extends Supertesting{  public void method() throws ArrayIndexOutOfBoundsException{  System.out.println("Inside method::Subclasstest");  }  }  What will be output of above program?   1. Inside method::Subclasstest 2. Inside method::Supertesting 3. RuntimeException is thrown 4. Compilation fails | 2 |
| Q21 | public class ExceptionTest {  public static void main(String...strings ){  Supertesting superobj = new Subclasstest();  superobj.method();  }  }  class Supertesting{  public void method() throws NullPointerException{  System.out.println("Inside method::Supertesting");  }  }  class Subclasstest extends Supertesting{  public void method() throws ArrayIndexOutOfBoundsException{  System.out.println("Inside method::Subclasstest");  }  }  What will be output of above program?   1. Inside method::Subclasstest 2. Inside method::Supertesting 3. RuntimeException is thrown 4. Compilation fails | 2 |
| Q22 | public class ExceptionTest {  public static void main(String...strings ){  Supertesting superobj = new Subclasstest();  superobj.method();  }  }  class Supertesting{  public void method() throws Exception{  System.out.println("Inside method::Supertesting");  }  }  class Subclasstest extends Supertesting{  public void method() throws NullPointerException{  System.out.println("Inside method::Subclasstest");  }  }  What will be output of above program?   1. Inside method::Subclasstest 2. Inside method::Supertesting 3. RuntimeException is thrown 4. Compilation fails | 2 |
| Q23 | public class FinalTest {  public static void main(String[] args){  System.out.println("Inside main method");  method(3);    }  static void method(int ...i){  System.out.println("Inside varargs version");  }      static void method(int i, int y){  System.out.println("Inside int version");  }    static void method(Integer ...integers){  System.out.println("Inside Integer version");  }  }  What will be output of above program?   1. Inside Integer version 2. Inside int version 3. Inside varargs version 4. Compilation fails | 2 |
| Q24 | public class SubStringTesting {  public static void main(String args[]){  String Str = new String("Welcome to Tutorialspoint.com");  System.out.print("Return Value :" );  Str = Str.substring(10);  System.out.println(Str );  Str.substring(10,15);  Str = Str.substring(3);  System.out.println(Str);  }  }  What will be output of above program? | 2 |
| Q25 | class Base{  public void move(){  System.out.println("Inside move::Super");  }    public void moveMethod(){  move();  }    int move;  }  class Subcls extends Base{  public void move(int i){  System.out.println("Inside move::Subclass");  moveMethod();  }  }  public class InheritanceTest {  public static void main(String[] args){  Subcls sub = new Subcls();  sub.move(2);  System.out.println("move value= "+(sub.move));  }  }  What will be output of above program? | 2 |