

Code 2- course no- Normal font on the cover page

Instruction: use the attached answer sheet for part-I and Part-IV question

Part I: Short answer question (2 pts each)

1. Briefly describe the following object oriented basic concepts data abstraction , data encapsulation and data binding –**give full mark at least two concept must be explained**
2. Explain what each term of “ **public static void main(String args[])**” method mean in Java? **0.5mark for each explanation**
3. Define Abstract Class in java and support your answer by example (**1 pts for definition & pts for example**)

Part II: Find out the error in the following program and rewrite it again by correcting the errors in the given box (2) each

#1. Find out the error and rewrite it again

```
class A
{ private int x=4;
int z=10;
static int y=10;
A()
{
return x+z;
}
private void display ()
{
System.out.println("sum="+ sum+ "y="+y);
}
class B extends A{
int d=0;
}
public static void main(String arg[])
{ B b1=new B();
b1.x=5;
b1.z=10;
b1.display();
}}
```

```
class A
{ private int x=4;
int z=10;
static int y=10;
A()
{
return x+z; // 1 mark for Identification & correction
}
private void display ()
{
System.out.println("sum="+ sum+ "y="+y);
}
class B extends A{
int d=0;
}
public static void main(String arg[])
{ B b1=new B();
b1.x=5; // 0.5mark
b1.z=10;
b1.display();//0.5 mark
}}
```

#2. Find out the error and rewrite it again

```
abstract class test
{
abstract void add(int x,int y);
abstract void mult( int x ,int y);
void square(int x)
{System.out.println(x*x)
}
class ImpTest extends test {
void add( int a, int b)
{ System.out.println(a*b) ; }
}
class check{
public static void main(String args[])
{
test t1=new test();
}
```

```
abstract class test
{
abstract void add(int x,int y);
abstract void mult( int x ,int y);
void square(int x)
{System.out.println(x*x)
}
class ImpTest extends test {
void add( int a, int b)
{ System.out.println(a*b) ; }
void mult( int x ,int y)
{ System.out.println(a*b) ; }
}
class check{
public static void main(String args[])
{
test t1=new test();//cannot create an object using abstract class
ImpTest t1=new ImpTest(); // 1 mark
}
```

Part: III: Assume all programs are error free and Write the output of the following program in the given box

#1. Write the output (3pts)

```
class Teststatic2 {
    static int x=8;
Teststatic2 () {
    ++x;
    incr(x);}
    static void incr(int x){System.out.println(x+7); }
static { int y=5;
System.out.println(++y);
}
    public static void main(String args[])
{
Teststatic a=new Teststatic();
System.out.println(++x);
}}
```

1 Mark for each output

6
16
10

#2. Write the output (3 pts)

```
class Area {
    int x=5, y=6;
    Area(){this(3);}
    Area(int x)
    {this(x,8);}
    Area(int a, int z)
    {x=a+3;y=z+4;
    System.out.println(x*y*5); } }
class Tri extends Area{
    Tri()
    {super(6);
    System.out.println(x*y*0.5);}
    Tri(int z)
    {super(z);} }
class Circle extends Tri{
    public Circle() {System.out.println(x+y); }
    public static void main(String args[])
{   Area a1=new Area();
    Tri t2=new Tri();
    Circle c1=new Circle(); } }
```

0.5 mark for each output

360
540
54.0
540
54.0
21

#3. Write the output (3pts)

```
class Test {
    int a,b;
    public Test(){
        send();
    }
    public Test(int z){
        this(4,z);}
    Test(int a, int x) {
        this.a = a;
        b=x;}
    Test mtest(Test t) {
        t.a+=3;
        t.b+=6;
    }
    Test temp = new Test(a,b);
    return temp; }
    void send(){
        mtest(this);}
}

class TestObjasretuntype {
    public static void main(String args[]) {
        Test ob1 = new Test();
        Test ob2= new Test(5,4);
        Test ob3=new Test(3);
        //ob1.send();
        ob1= ob1.mtest(ob1);
        ob2=ob2.mtest(ob2);
        ob3=ob3.mtest(ob3);
        System.out.println("After first call value ob1.b: " + ob1.b);
        System.out.println("After first call value ob2.b: " + ob2.b);
        System.out.println("After first call value ob3.b: " + ob3.b);
        ob1=ob1.mtest(ob1);
        ob2 = ob2.mtest(ob2);
        ob3=ob3.mtest(ob3);
        System.out.println("ob1.a , ob2.a and ob3.a. after second Call:" + ob1.a+" "+ob2.a+" "+ob3.a);
    }
}
```

0.5 mark for each output

After first call value ob1.b: 12

After first call value ob2.b: 10

After first call value ob3.b: 9

ob1.a , ob2.a and ob3.a. after second Call: 9 11 10

#4. Write the output (3 pts)

```
public class rectangle {
    double width;
    double length;
    rectangle()    {
        width=4;
        length =6;
        calcArea(width,length);
    }
    rectangle(double width,double length)
    {
        this.width=width;
        this.length= length;
        width=length=7;
        calcArea(width,length);
        calcArea(this.width,this.length);
        this.width=length=2;
        calcArea(width,length);
        calcArea(this.width,this.length);
    }
    void calcArea(double width,double length){
        double ar= width*length;
        System.out.println("AREA="+ ar);
    }
}

class RectDemo {
    public static void main(String args[]){
        rectangle rect1 = new rectangle();
        rect1.calcArea(9,4);
        rectangle rect2 = new rectangle(8,6);
    }
}
```

0.5 mark for each output

AREA=24.0

AREA=36.0

AREA=49.0

AREA=48.0

AREA=14.0

AREA=12.0

Part IV: Program Writing (8%)

Assume ASTU students, students are categorized into extension and regular student, regular students are categorized into undergraduate and graduate students, the behavior and state of all classes listed in the table below

| Class | Behavior | state |
|-----------------------|--|--|
| Extension student | Registration, add and drop course , show grade, make payment | studId,name, dept, semester payment, |
| Regular student | Registration ,add and drop course , view grade, | studId ,name, dept, dormitory no,blockno,café type |
| Undergraduate student | Registration ,add and drop course , view grade, fill-CostSharing | studId ,name, dept, dormitory no, blockno , café type ,costsharing |
| Graduate student | Registration ,add and drop course , view grade, thesis payment, submit sponsor | studId ,name, dept, thesis fund, specialization ,sponsor |

Based on the above case attempt the following questions

1. Based on the class behavior and state in the above case ,Define all classes by implementing the concepts of inheritance using java (4pts)
2. Implement either registration or Add&Drop course Behavior (2pts)
3. Identify the type of inheritance that can be implemented in the above case support your answer by example (2pts)

Answer : #1 each class has 1 mark

```
class student{
String studId, name, dept
void Registration(String ID, String name, String dept)
{
studId=ID;
this.name=name;
this.dept=dept;
}
void addDrop( )
{}
void view ()
{}}
```

#2 –

2marks

```

class extensionStudent extends student
{
    float payment;
    void makepayment(float pay){
        payment=pay;
    }
}

class RegularStudent
{
    int dormitory no,
    String blockno,
    Sting café type
}

class UnderGraduate extends RegularStudent
{
    float costsharing;
    void fillCostSharing(){ }
}

class PostGraduate extends RegularStudent
{
    Stiring Specialization ;
    float thesis-fund;
    String sponsor ;
    void makepayment(){ }
    void sponsor (){}
}

```

#3 1 mark for each

A) Multiple : Student → RegularStudent → UnderGraduate

B) Hirarcical :

