**(6 points)** What is the output?

class A

{

public A() {System.out.println(1);}

}

class B extends A

{

public B() {System.out.println(2);}

}

class C extends B

{

public C() {System.out.println(3);}

}

public class MainClass

{

public static void main(String[] args)

{

C c = new C();

}

}

ANS:

1

2

3

*For the 3 questions below, consider the following class definition:*

public class AClass

{

private int x;

private int y;

public AClass(int a, int b)

{

x = a;

y = b;

}

public int addEm( )

{

return x + y;

}

public void changeEm( )

{

x++;

y--;

}

public String toString( )

{

return "" + x + " " + y;

}

}

2. (4 points)Consider that you want to extend AClass to BClass. BClass will have a third int instance data, z. Which of the following would best define BClass'constructor?

A) public BClass(int a, int b, int c)

{

super(a, b, c);

}

B) public BClass(int a, int b, int c)

{

x = a;

y = b;

z = c;

}

C) public BClass(int a, int b, int c)

{

z = c;

}

D) public BClass(int a, int b, int c)

{

super(a, b);

z = c;

}

E) public BClass(int a, int b, int c)

{

super( );

}

Answer: D

3.(5 points)In BClass, rewrite addEm method now to add all three values and return the sum in BClass.

1. public void addEm( )  
    {  
    return z + super.addEm();  
    }
2. public int addEm( )  
    {  
    return x + y + z;  
    }
3. public int addEm( )  
    {  
    return z + super.addEm();  
    }
4. public void addEm( )  
    {  
    return x + y + z;  
    }

ANS: C

4. (4 points)Which of the following would best redefine the toString method for BClass?

A) public String toString(int z)

{

return " " + x + " " + y + " " + z;

}

B) public String toString( )

{

return super.toString( );

}

C) public String toString( )

{

return super.toString( ) + " " + z;

}

D) public String toString( )

{

return super.toString( ) + " " x + " " + y + " " + z;

}

E) public String toString( )

{

return " " + x + " + y + " " + z;

}

Answer: C

*For the 3 questions below, assume that Student, Employee and Retired are all extended classes of Person, and all four classes have different implementations of the method getMoney. Consider the following code where ... are the required parameters for the constructors:*

Person p = new Person(...);

int m1 = p.getMoney( ); // assignment 1

p = new Student(...);

int m2 = p.getMoney( ); // assignment 2

if (m2 < 100000) p = new Employee(...);

else if (m1 > 50000) p = new Retired(...);

int m3 = p.getMoney( ); // assignment 3

5. (4 points)The reference to getMoney( ) in assignment 1 is to the class

A) Person

B) Student

C) Employee

D) Retired

E) none of the above, this cannot be determined by examining the code

Answer: A

Explanation: A) At this point of the program, p is a Person, and so getMoney is a reference to Person's getMoney method.

6. (4 points)The reference to getMoney( ) in assignment 2 is to the class

A) Person

B) Student

C) Employee

D) Retired

E) none of the above, this cannot be determined by examining the code

Answer: B

Explanation: B) At this point of the program, p is a Student, and so getMoney is a reference to Student's getMoney method.

7. (4 points) The reference to getMoney( ) in assignment 3 is to the class

A) Person

B) Student

C) Employee

D) Retired

E) none of the above, this cannot be determined by examining the code

Answer: E

*For the next 2 questions, consider the following class definition:*

public class AnyClass

{

private int x;

public AnyClass(int newValue)

{

x = newValue;

}

}

8. (4 points)Which of the following is true about the class AnyClass?

A) It has no parent class

B) Its parent class is Object

C) Its parent class is Java

D) It can not be extended

E) It has a default child called Object

Answer: B

9. (4 points)If q1 and q2 are objects of AnyClass, then q1.equals(q2)

A) is a syntax error since equals is not defined in the class

B) is true if q1 and q2 both store the same value of x

C) is true if q1 and q2 reference the same AnyClass object

D) is never true

E) throws a NullPointerException

Answer: C

10. (6 points)What is printed by the following code?

{

class Figure

{

void display( )

{

System.out.println("Figure");

}

}

class Rectangle extends Figure

{

void display( )

{

System.out.println("Rectangle");

}

}

class Box extends Figure

{

void display( )

{

System.out.println("Box");

}

}

public static void main(String[ ] args)

{

Figure f = new Figure( );

Rectangle r = new Rectangle( );

Box b = new Box( );

f.display( );

f = r;

f.display( );

f = b;

f.display( );

}

}

ANS:

Figure

Rectangle

Box

11. (5 points) This program doesn’t compile, explain why.

abstract public class Demo  
{  
 public int a;  
 Demo()  
 {  
 a = 10;  
 }  
 abstract public void set();  
 abstract public void get();  
}  
public class Test extends Demo  
{  
 public void set(int a)  
 {  
 this.a = a;  
 }  
 final public void get()  
 {  
 System.out.println("a= " + a);  
 }  
 public static void main(String[] args)  
 {  
 Test obj = new Test();  
 obj.set(20);  
 obj.get();  
 }  
}

ANS

Test is not abstract and does not override abstract method set() in Demo

12.(5 points)What is the output?

public class SuperDemo{

public void show(){

System.out.println("super class method called");

}

}

public class SubDemo extends SuperDemo{

public void show(){

System.out.println("sub class method called");

}

 public static void main(String args[]){

SubDemo subobj=new SubDemo();

subobj.show();

}

ANS: sub class method called

13. (5 points) What is the output?

public abstract class One {  
  
 private int x = 19;  
 protected int y = 123;  
  
 public abstract void print();  
  
 public void show() {  
 System.out.println(x);  
 }   
}

public class Two extends One   
{  
  
 private int z = -45;  
  
 public void print() {  
 System.out.println(z);  
 System.out.println(y);  
 }   
}

public class Main   
{  
 public static void main(String [] args)   
 {  
 One it = new Two();  
 it.print();  
 it.show();  
 }   
}   
  
ANS:

-45

123

19

14. **(5 points) Mike has written the code like below. But, it is showing compile time error. Can you identify what mistake he has done?**

class X

{

//class X members

}

class Y

{

//class Y members

}

Class Z extends X, Y

{

//class Z members

}

ANS:

In Java, a class can not extend more than one class. Class Z is extending two classes – Class X and Class Y. It is a compile time error in java.

15. Suppose the class Employee is declared as follows:

public class Employee  
{  
 private String name;  
 private double baseSalary;  
   
 public void setName(String newName){...}  
 public void setBasSalary(double newSalary){...}  
 public String getName(){...}  
 public double getSalary(){...}  
}

a)(4 points)Declare a class Manager that inherits from the class Employee and adds an instance variable bonus for storing a salary bonus. Omit constructors and methods.

ANS:

public class Manager extends Employee  
{  
 private double bonus;  
}

b) (4 points)Which instance variables does the Manager class have?

ANS: name, baseSalary, and bonus

c) (5 points)In the manager class, provide the method overrides the getSalary method from the class Employee so that it returns the sum of the salary and the bonus.

ANS: public double getSalary(){return super.getSalary() + bonus;}

Or public double getSalary(){return getSalary() + bonus;}

d) (5 points) Based on the above question, which methods does the manager class inherit?

ANS: getName, setName, setBaseSalary

e) (5 points)In the manager class, override the getName method so that managers have a \* before their name (such as \*John Smith).

ANS: public String getName(){ return “\*” + super.getName();}

Or public String getName(){ return "\*" + getName();}

16. (8 points) For the following four statements, write “ok” for valid statements and write “not ok” for invalid statements. ? (Answer each of them. It is not a multiple choice question)

public abstract class Account  
{  
 public abstract void deductFees();  
 ...  
}  
public class SavingsAccount extends Account   
{  
 ...  
 public void deductFees()  
 {  
 ...  
 }  
}

1. Account anAccount;
2. Account anAccount = new Account();
3. Account anAccount = new SavingsAccount();
4. Account anAccount = null;

ANS: Ok, not ok, ok, ok

17. (4 points) Suppose the class Sub extends the class Sandwich. Which assignment is legal and which statement is not legal? (Answer each of them. It is not a multiple choice question)

Sandwich x = new Sandwich();

Sub y = new Sub();

1. x = y;
2. y = x;
3. y = new Sandwich();
4. x = new Sub();

ANS: legal, not , not, legal