

Chapter 3 – Classes & Objects

Question 1:

Given the code below,

```
1. public class Try
2. {
3.     private static int sum = 20;
4.     public static void main(String args[])
5.     {
6.         Try t1 = new Try();
7.         Try.sum++;
8.         Try t2 = new Try();
9.         t2.sum++;
10.        t1=new Try();
11.        t1.sum++;
12.        sum--;
13.        new Try().sum++;
14.        t1.sum++;
15.        System.out.println(sum);
16.    }
17.}
```

The output is:

- A) 24
- B) 22
- C) 20
- D) The program doesn't compile
- E) 21

Question 2:

Given the code below,

```
1.class StaticStuff
2.{
3.    static int x = 40;
4.    static
5.    {
6.        x += 20;
7.    }
8.    public static void main(String args[])
9.    {
10.        System.out.println("x = " + x);
11.    }
12.    static
13.    {
14.        x /= 15;
15.    }
16.}
```

The output is:

- A) 4
- B) 40
- C) 60

- D) The program doesn't compile
- E) 8

Question 3:

Given the code below,

```
1.class Robin
2.{
3.    void verMethod(int num) {System.out.println("int version");}
4.    void verMethod(float num) {System.out.println("float version");}
5.    public static void main(String args[])
6.    {
7.        Robin rob=new Robin();
8.        char ch = 'a';
9.        rob.verMethod(ch);
10.    }
11.}
```

The output is:

- A) int version
- B) float version
- C) The code will compile but will throw an exception
- D) The code will not compile because there is no version of verMethod that receives a char argument

- () E) The code will not compile because methods can not be overridden in the same class

Question 4:

Which of the statements below is true ?

- A) Object references can be converted in both method calls and assignments
- B) Object references cannot be converted
- C) Object references can be converted only in assignments
- D) Object references can be converted only in method calls
- E) Only primitive type variables can be converted

Question 5:

Which of the statements below is true ?

- A) Both primitives and object references can be both converted and cast
- B) Only primitive types are converted automatically. To change the type of an object reference, you have to do casting.

Question 6:

Given the code below,

```
1.class StaticStuff
2.{
3.    public static void main(String args[])
4.    {
5.        Object obj = new Float();
6.        String [] names = new String[100];
7.        Float floty = new Float(1.5f);
8.        obj = names;
9.        floty = obj;
10.    }
11.}
```

Which of the following lines are true:

- A) line 9 will not compile
- B) line 8 will not compile
- C) line 7 will not compile
- D) line 6 will not compile
- E) The code will compile successfully
- F) line 5 will not compile

Question 7:

Given the code below,

```
1. public class Derive1
2. {
3.     public Derive1(float j)
4.     {
5.     }
6.     public Derive1(int j, int k)
7.     {
8.
9.     }
10.}
```

Which of the following are legitimate calls to construct instances of the Derive1 class?

- A) Derive1 d = new Derive1(1);
- B) Derive1 d = new Derive1("one");
- C) Derive1 d = new Derive1(1,3,4);
- D) Derive1 d = new Derive1(4,3);
- E) Derive1 d = new Derive();

Question 8:

Consider the following class definition,

```
1. public class Test1
2. {
3.     public float method(float a, float b)
4.     {
5.     }
6.
7. }
```

Which of the following methods can be legally added (individually) at line 6 ?

- A) public float method (float c, float d) { }
- B) public float method (float a, float b, float c) { }
- C) public void method (int a, int b) { }
- D) public int method (int a) { }
- E) public float method(float a, float b) { }

Question 9:

Which of the following statements are true ?

- A) It is possible to declare more than one class in the same file
- B) It isn't possible to declare more than one class in the same file
- C) The name of the source file must be identical to the name of the class that its declaration is placed in that file only if the file has one class declaration in it
- D) If a source file has more than one class declaration in it than its name must be identical to the name of the public class
- E) Each class must be declared in a file, which its name is identical to the name of the class.

Question 10:

Consider the following class definition,

```

1. public class Try
2. {
3.     public float method (float a, float b)
4.     {
5.     }
6.
7. }
```

Which of the following methods can be legally added (individually) at line 6 ?

- A) public int method (float a, float b) {}
- B) public float method (int a, int b) {}
- C) public float method (float a) {} throws Exception
- D) public float method (float num1, float num2) {}
- E) public float method(float a, float b) {}

Question 11:

Forcing garbage collection of an object

- A) can be done by calling the System.gc() method
- B) can be done by calling the Runtime.gc() method
- C) can be done by setting all the variables that contain a reference to the object to NULL
- D) is not possible
- E) can be done by calling the Toolkit.gc() method

Question 12:

Consider the following application:

```
1.class Question4
2.{
3.    public static void main(String args[])
4.    {
5.        Zombit z = new Zombit(12);
6.        z.jump(2);
7.        System.out.println("z value is : " + z.value);
8.    }
9.}
10.
11. public class Zombit
12. {
13.    public int value;
14.    public Zombit(int val)
15.    {
16.        value = val;
```

```
17. }
18. public void jump(int val)
19. {
20.     value+= val;
21. }
22. }
```

What value is printed out at line 7 ?

- () A) 14
- () B) 12
- () C) 10
- () D) 8
- () E) 4

Question 13:

Consider the following application:

```
1. class Question13
2. {
3.     public static void main(String args[])
4.     {
5.         Harmony h = new Harmony();
6.         h.held = 1;
7.         h.jump(h);
8.         System.out.println(h.held);
9.     }
10. }
11.
12. class Harmony
13. {
```

```
14. public int held;
15. public void jump(Harmony theHarmony)
16. {
17.     theHarmony.held++;
18. }
19. }
```

What value is printed out at line 8?

- A) 2
- B) 3
- C) 11
- D) 100
- E) 1

Question 14:

Consider the following application:

```
1. class Question14
2. {
3.     public static void main(String args[])
4.     {
5.         Harmony h = new Harmony();
6.         h.held = 1;
7.         h.jump(h.held);
8.         System.out.println(h.held);
9.     }
10. }
```

```
11.  
12. class Harmony  
13. {  
14.     public int held;  
15.     public void jump(int val)  
16.     {  
17.         val++;  
18.     }  
19. }
```

What value is printed out at line 8?

- A) 2
- B) 3
- C) 11
- D) 100
- E) 1

Question 15:

Consider the following application:

```
1.class Question15  
2.{  
3.     public static void main(String args[])  
4.     {  
5.         int number = 24.2;  
6.         Car myCar = new Car();
```

```
7.     myCar.jump(number);
8.     myCar.jump(myCar.size);
9.     System.out.println(number+myCar.size);
10.  }
11. }
```

```
1. class Car
2. {
3.     int size = 0.8;
4.     public void jump(int num)
5.     {
6.         num++;
7.     }
8. }
```

What value is printed out at line 9?

- A) The code won't compile successfully
- B) 25.8
- C) 50
- D) 49.2
- E) 26.6

Question 16:

Consider the following code:

1. Object o = new Object();
2. Float f = new Float(15.5);
3. f = o;

Which of the statements below is true ?

- A) line 3 won't compile
- B) the code will successfully compile, but won't run

Question 17:

Consider the following code:

1. class Question
2. {
3. public Question(int index, int num)
4. {
5. }
6. public Question(int index)
7. {
8. }
9. }

Which of the following are legitimate instantiating ?

- A) Question a = new Question();
- B) Question a = new Question(3.2);
- C) Question a = new Question(4,2);
- D) Question a = new Question(8);
- E) Question a = new Question("fff");

Question 18:

True or False:

It is possible to subclass the java.lang.Math class.

- A) true
- B) false

Question 19:

True or False:

It is possible to instantiate the java.lang.Math class.

- A) true
- B) false

Question 20:

Given the code below,

```
1. public float getMagicNumber(float number)
2. {
3.     int result;
4.     if (number<0)
5.         result = -number;
6.     return result;
7. }
```

- A) The code will compile successfully, and executes without any exception.

- B) The code will compile successfully but an exception will be thrown during its execution.
- C) Initializing of the result variable will solve the compilation problem.

Question 21:

True or False:

The native modifier refers only to methods, and it indicates that the body of the method lies outside of the Java Virtual Machine, in a library. Native methods are usually written in C/C++.

- A) True
- B) False

Question 22:

True or False:

Using the keyword "this" inside a constructor in order to call another constructor of the same class must be placed as the first statement of the constructor.

- A) True
- B) False

Question 23:

Which of the following statements is true (one or more) :

- A) A collection doesn't have a special order
- B) A collection can have the same item more than once
- C) A list can have the same item more than once, and its items are ordered
- D) A set rejects duplicates
- E) A map supports the using of a key field

Question 24:

True or False:

While instance variables are initialized automatically, local variables must be initialized manually before use. Compilation of a code fragment, that has a condition where local variables might be used before being initialized, fails.

- A) True
- B) False

Question 25:

Given the following method code:

```
public void getLuckyNumber(int z)
{
    int x;
    if (z>1000)
        x = 120;
```

```
System.out.println("The Lucky number is : " + x);  
}
```

which of the following statements (one or more) is true ?

- A) The output depends in z.
- B) If z equals 2000 then the output will be 120.
- C) If z equals 1000 then an exception will be thrown.
- D) If z equals 500 then the output will be 0.
- E) The compilation of this code fragment will fail.

Question 26:

The Collection API contains three main interfaces: Collection, Set and List. These interfaces are implemented in several classes which are usually called : collections.

Each of these collections is a single object that represents a group of other objects. The collections maintain references to objects of type Object. In this way any object can be stored in a collection. Which of the following sentences is true ?

- A) The classes that implement the Collection interface represent a group of objects without any specific order.
- B) The classes that implement the Set interface represent a group of objects with no duplication.
- C) The classes that implement the List interface represent a group of objects that allows duplications.

Question 27:

Given the code below,

```
1. public class Try
2. {
3.     private static int sum;
4.     public static void main(String args[])
5.     {
6.         Try t1 = new Try();
7.         Try.sum++;
8.         Try t2 = new Try();
9.         t2.sum++;
10.        t1=new Try();
11.        t1.sum--;
12.        t1.sum--;
13.        System.out.println(sum);
14.    }
15.}
```

The output is:

- A) 0
- B) 22
- C) 20
- D) The program doesn't compile
- E) 21

Question 28:

Given the code below,

```
1.class StaticStuff
2.{
3.    static int x;
4.    static
5.    {
6.        x += 20;
7.    }
8.    public static void main(String args[])
9.    {
10.        System.out.println("x = " + x);
11.    }
12.    static
13.    {
14.        x /= 16;
15.    }
16.}
```

The output is:

- A) 1
- B) 1.33333333
- C) 1.25
- D) The program doesn't compile
- E) 1.2

Question 29:

True or False:

Forcing garbage collection of an object can be done by calling the System.gc() method

- A) True
- B) False

Question 30:

True or False:

Using the keyword "this" inside a constructor in order to call another constructor of the same class can be placed as the last statement of the constructor.

- A) True
- B) False

Question 31:

Consider the following application:

```
1.class Question4
2.{
3.    public static void main(String args[])
4.    {
5.        Zombit z = new Zombit(12);
6.        z.jump(z.value);
```

```
7.         System.out.println("z value is : " + z.value);
8.     }
9. }
10.
11. public class Zombit
12. {
13.     public int value, result;
14.     public Zombit(int val)
15.     {
16.         result = val;
17.     }
18.     public void jump(int val)
19.     {
20.         value+= val;
21.     }
22. }
```

What value is printed out at line 7?

- A) 14
- B) 12
- C) 10
- D) 0
- E) 4

Question 32:

Consider the following code:

```
1. class Question
2. {
3.     public Question(int index, int num)
4.     {
```

```
5.  }
6.  public Question(int index)
7.  {
8.  }
9. }
```

Which of the following are legitimate instantiating ?

- A) Question a = new Question();
- B) Question a = new Question(3.2);
- C) Question a = new Question(4,2);
- D) Question a = new Question(8);
- E) Question a = new Question('s');

Question 33:

Consider the following code:

1. Object o = new String();
2. Float f = new Float();
3. f = o;

Which of the statements below is true ?

- A) line 3 won't compile
- B) the code will successfully compile, but won't run

Question 34:

True or False:

Each class must be declared in a file which its name is identical to the name of the class.

- A) True
- B) False

Question 35:

True or False:

Java allows declaring a global variable (like C\C++) that doesn't belong to a class.

- A) True
- B) False

Question 36:

Given the code below:

```
1.      class Zombit
2.      {
3.          static int bomba;
4.          public static void main(String args[])
5.          {
6.              Zombit zombit1 = new Zombit(2);
7.              Zombit zombit2 = new Zombit(3);
8.              Zombit zombit3 = new Zombit();
```



```
9.          System.out.println("bomba equals to : " + bomba);
10.         }
11.         Zombit(int number)
12.         {
13.             bomba += number;
14.         }
15.     }
```

The output is:

- A) bomba equals to 5
- B) bomba equals to 3
- C) bomba equals to 0
- D) The code doesn't compile successfully

Question 37:

Given the following code:

```
1.         class Pompa
2.         {
3.             static int counter;
4.             Pompa()
5.             {
6.                 counter++;
7.             }
8.             protected void finalize()
9.             {
10.                counter--;
11.            }
12.            public static void main(String args[])
13.            {
```

```

14.          Pompa pompa1 = new Pompa();
15.          Pompa pompa2 = new Pompa();
16.          Pompa pompa3 = new Pompa();
17.          Pompa pompa4 = new Pompa();
18.          Pompa pompa5 = new Pompa();
19.          pompa5 = null;
20.          System.out.println("counter equals " + counter);
21.          }
22.          }

```

- A) The output is: counter equals 5
- B) The output is: counter equals 4
- C) The output is: counter equals 0
- D) The output might be: counter equals 4
- E) The output might be: counter equals 5

Question 38:

Given the following code:

```

1.      class Chompy
2.      {
3.          private int id;
4.          private Chompy next;
5.          Chompy()
6.          {
7.              next = null;
8.              id = 0;
9.          }
10.         Chompy(int id)
11.         {
12.             next = null;

```

```
13.         this.id = id;
14.     }
15.     Chompy(int id, Chompy next)
16.     {
17.         this.next = next;
18.         this.id = id;
19.     }
20.     public static void main(String args[])
21.     {
22.         Chompy first = new Chompy(101, new Chompy(102));
23.         System.out.println(first.next.id);
24.     }
25. }
```

The output is:

- A) 0
- B) 101
- C) 102
- D) The code doesn't compile successfully

Question 39:

Given the code below:

```
1.     public class Rectangle
2.     {
3.         double x;
4.         double y;
5.         public static void main(String args[])
6.         {
7.             Rectangle recy1 = new Rectangle(2,1);
```

```
8.         Rectangle recy2 = new Rectangle(1,5);
9.         Rectangle recy3 = new Rectangle(2,3);
10.        Rectangle recy4 = new Rectangle(2,2);
11.        double sum = 0;
12.        sum += recy1.area();
13.        sum += recy2.area();
14.        sum += recy3.area();
15.        sum += recy4.area();
16.        System.out.println(sum);
17.    }
18.    public double area()
19.    {
20.        return x*y;
21.    }
22.    public Rectangle(double x,double y)
23.    {
24.        this.x = x;
25.        this.y = y;
26.    }
27.    }
```

The output is :

- A) 14.0
- B) 14
- C) 15
- D) 0
- E) 17
- F) 17.0

Question 40:

Given the code below:

```
1. class Plomba
2. {
3.     int id;
4.     private Plomba()
5.     {
6.         id = 0;
7.     }
8.     private Plomba(int id)
9.     {
10.        this.id = id;
11.    }
12. }
```

```
1. class App
2. {
3.     public static void main(String args[])
4.     {
5.         Plomba plomby1 = new Plomba();
6.         Plomba plomby2 = new Plomba(72);
7.         System.out.println(plomby1.id + plomby2.id);
8.     }
9. }
```

The output is:

- A) 072
- B) 72
- C) 0
- D) 720
- E) The code doesn't compile successfully

Question 41:

Given the following code:

```
1. class Tambur
2. {
3.     int num;
4.     public static void main(String args[])
5.     {
6.         num = 10;
7.         num = qualala(num);
8.         System.out.println("num="+num);
9.     }
10.    int qualala(int number)
11.    {
12.        return number * 10;
13.    }
14. }
```

The output is :

- A) 10
- B) 100
- C) 1000
- D) The code doesn't compile successfully

Question 42:

Given the code below:

```
1.      class Rublika
2.      {
3.          double xCor;
4.          double xCor;
5.          double width;
6.          double height;
7.          Rublika sun;
8.          Rublika(double xCor, double yCor, double width, double height)
9.          {
10.             this.xCor = xCor;
11.             this.yCor = yCor;
12.             this.width = width;
13.             this.height = height;
14.         }
15.         public double getArea()
16.         {
17.             if (sun==null)
18.                 return width*height;
19.             else
20.                 return sun.getArea()+width*height;
21.         }
22.         void connect(Rublika theSun)
23.         {
24.             sun = theSun;
25.         }
26.         public static void main(String args[])
27.         {
28.             Rublika first, second;
29.             first = new Rublika(2,3,10,20);
```

```
30.         second = new Rublika(12,14,5,4);
31.         first.connect(second);
32.         double sum = first.getArea() + second.getArea();
33.         System.out.println(sum);
34.     }
35. }
```

The output is :

- A) 200
- B) 220
- C) 240
- D) 440
- E) 460
- F) 480

Question 43:

Given the code below:

```
1. class Duba
2. {
3.     double weight;
4.     double age;
5.     Duba child1, child2;
6.     Duba(double weight, double age)
7.     {
8.         this.weight = weight;
9.         this.age = age;
10.    }
11.    public boolean isOk()
12.    {
```



```
13.         if ((weight/age)>5)
14.             return false;
15.         else
16.             return true;
17.     }
18.     public boolean isAllOk()
19.     {
20.         if (child1==null && child2==null)
21.             return isOk();
22.         if (child1==null)
23.             return (isOk() && child2.isOk());
24.         if (child2==null)
25.             return (isOk() && child1.isOk());
26.         return (isOk() && child1.isOk() && child2.isOk());
27.     }
28.     public void connectChild1(Duba other)
29.     {
30.         if (child1==null)
31.             child1=other;
32.     }
33.     public void connectChild2(Duba other)
34.     {
35.         if (child2==null)
36.             child2=other;
37.     }
38.     public static void main(String args[])
39.     {
40.         Duba duba1 = new Duba(100,20);
41.         duba1.connectChild1(new Duba(100,19));
42.         duba1.connectChild2(new Duba(80,4));
43.         if (duba1.isOk() && !duba1.isAllOk())
44.             System.out.println("State A");
45.     }
```

46. }

The output will be: "State A".

- A) True
- B) False

Question 44:

Given the code below:

```
1. private class Tropez
2. {
3.     int id;
4.     public static void main(String args[])
5.     {
6.         Tropez tropy = new Tropez();
7.         tropy.id = 99;
8.         tropy.id++;
9.         System.out.println(++tropy.id);
10.    }
11. }
```

The output is:

- A) The code doesn't compile successfully
- B) 99
- C) 0
- D) 100
- E) 101

Question 45:

Given the code below:

```
1.  class Gorky
2.  {
3.      private int theNumber;
4.      int moveTheNumber(int steps)
5.      {
6.          int result;
7.          result = theNumber >> steps;
8.          return result;
9.      }
10.     public static void main(String args[])
11.     {
12.         Gorky gorky = new Gorky();
13.         gorky.theNumber = -5;
14.         gorky.moveTheNumber(32);
15.         System.out.println(gorky.theNumber);
16.     }
17. }
```

The output is:

- A) -5
- B) 0
- C) The code doesn't compile successfully
- D) -1
- E) The compile successfully but the output depends on the platform on which the program is activated

Question 46:

Given the code below:

```
1.  public class BitMove
2.  {
3.      private int num;
4.      BitMove(int number)
5.      {
6.          num = number;
7.      }
8.      public int getNum()
9.      {
10.         return num;
11.     }
12.     public void moveTheBits(int steps)
13.     {
14.         num = num << steps;
15.     }
16.     public void moveTheBits(int steps, int times)
17.     {
18.         for (int i=0; i<times; i++)
19.             num = num << steps;
20.     }
21.     public static void main(String args[])
22.     {
23.         BitMove bitMove1 = new BitMove(1);
24.         BitMove bitMove2 = new BitMove(1);
25.         bitMove1.moveTheBits(64);
26.         bitMove2.moveTheBits(1,32);
27.         System.out.println("bitMove1 number is : " + bitMove1.getNum());
28.         System.out.println("bitMove2 number is : " + bitMove2.getNum());
29.     }
```

30. }

The output is:

- () A) bitMove1 number is 1
bitMove2 number is 0
- () B) bitMove1 number is 0
bitMove2 number is 1
- () C) bitMove1 number is 0
bitMove2 number is 0
- () D) bitMove1 number is 1
bitMove2 number is 1
- () E) The code doesn't compile
- () F) None of the above is true

Question 47:

Given the code below :

```
1. class MainObject
2. {
3.     Other other;
4.     int luckyNumber = 10;

5.     public MainObject(MainObject mainObject)
6.     {
7.         other = mainObject.other;
8.         luckyNumber = mainObject.luckyNumber;
9.     }
```

```
10.     public MainObject(Other theOther)
11.     {
12.         other = theOther;
13.     }

14.     public MainObject()
15.     {
16.         other = null;
17.     }

18.     public void update()
19.     {
20.         if(other==null)
21.             luckyNumber = 9;
22.         else
23.             other.change(this);
24.     }

25.     public void setLuckyNumber(int num)
26.     {
27.         luckyNumber = num;
28.     }

29.     public int getLuckyNumber()
30.     {
31.         return luckyNumber;
32.     }

33.     public static void main(String args[])
34.     {
35.         Other other = new Other(7);
36.         MainObject mainObject1 = new MainObject(other);
37.         mainObject1.update();
```

```
38.         MainObject mainObject2 = new MainObject(other);
39.         mainObject2.update();
40.         MainObject mainObject3 = new MainObject();
41.         mainObject3.update();
42.         MainObject mainObject4 = new MainObject(mainObject3);
43.         mainObject4.update();
44.         System.out.println(mainObject1.getLuckyNumber() + ","
45.             + mainObject2.getLuckyNumber() + ","
46.             + mainObject3.getLuckyNumber() + ","
47.             + mainObject4.getLuckyNumber());
48.     }
49. }
```

```
50. class Other
51. {
52.     public void change(MainObject mainObj)
53.     {
54.         mainObj.setLuckyNumber(8);
55.         mainObj.other = null;
56.     }
57.     public Other(int num){}
58. }
```

The output is :

- () A) 8,8,9,9
- () B) 7,8,9,10
- () C) 7,7,8,8
- () D) 10,9,8,7
- () E) The code compiles successfully but the output isn't one of the above answers.

Question 48:

Given the code below,

```
1. public class Try
2. {
3.     private static int sum = 20;
4.     public static void main(String args[])
5.     {
6.         Try t1 = new Try();
7.         Try.sum++;
8.         Try t2 = new Try();
9.         t2.sum++;
10.        t1=new Try();
11.        t1.sum--;
12.        Try.sum--;
13.        System.out.println(sum);
14.    }
15.}
```

- A) 19
- B) 22
- C) 20
- D) The program doesn't compile
- E) 21

Question 49:

Given the code below:

```
1. class Plomba
2. {
3.     private int id;
4.     private Plomba()
5.     { id = 0; }
6.     private Plomba(int id)
7.     { this.id = id; }
8. }
9.
10. class App
11. {
12.     public static void main(String args[])
13.     {
14.         Plomba plomby1 = new Plomba();
15.         Plomba plomby2 = new Plomba(72);
16.         System.out.println(plomby1.id + plomby2.id);
17.     }
18. }
```

The output is:

- A) 072
- B) 72
- C) 0
- D) 720
- E) The code doesn't compile successfully

Question 50:

Consider the following application:

```
1. public class Question58
2. {
3.     public static void main(String args[])
4.     {
5.         Zombit z = new Zombit(12);
6.         int number = 99;
7.         z.jump(number);
8.         System.out.println("number : " + number);
9.     }
10. }
11. class Zombit
12. {
13.     public int value;
14.     public Zombit(int val)
15.     {
16.         value = val;
17.     }
18.     public void jump(int val)
19.     {
20.         val++;
21.     }
22. }
```

What value is printed out at line 9?

- A) 99
- B) 100
- C) 0
- D) 98
- E) 12

Question 51:

Consider the following code:

```
1. class Question
2. {
3.     public Question(float index)
4.     {
5.     }
6.     public Question(int index)
7.     {
8.     }
9. }
```

Which of the following are legitimate instantiating ?

- A) Question a = new Question();
- B) Question a = new Question(3.2);
- C) Question a = new Question(4,2);
- D) Question a = new Question(8);
- E) Question a = new Question("fff");

Question 52:

Given the code below:

```
16.     class Zombit
17.     {
18.         static int bomba;
19.         public static void main(String args[])
20.         {
21.             Zombit zombit1 = new Zombit(2);
22.             Zombit zombit2 = new Zombit(3);
23.             Zombit zombit3 = new Zombit(2);
```

```
24.           System.out.println("bomba equals to : " + bomba);
25.         }
26.         Zombit(int number)
27.         {
28.             bomba += number;
29.         }
30.     }
```

The output is:

- () A) bomba equals to 5
- () B) bomba equals to 3
- () C) bomba equals to 0
- () D) bomba equals to 7

Question 53:

Given the code below:

```
25. class Plomba
26. {
27.     int id;
28.     Plomba()
29.     {
30.         id = 0;
31.     }
32.     Plomba(int id)
33.     {
34.         this.id = id;
35.     }
36. }
```

```
19. class App
20. {
```

```
21.     public static void main(String args[])
22.     {
23.         Plomba plomby1 = new Plomba();
24.         Plomba plomby2 = new Plomba(72);
25.         System.out.println((plomby1.id + plomby2.id));
26.     }
27. }
```

The output is:

- A) 072
- B) 72
- C) 0
- D) 720
- E) The code doesn't compile successfully

Question 54:

Consider the following application:

```
1.class Question62
2.{
3.     public static void main(String args[])
4.     {
5.         Zombit z = new Zombit(1);
6.         int number = 2;
7.         z.jump(number);
8.         System.out.println(z.value);
9.     }
10. }
```

```
11. public class Zombit
12. {
13.     public int value;
14.     public Zombit(int val)
15.     {
16.         value = val;
17.     }
18.     public void jump(int val)
19.     {
20.         value += val;
21.     }
22. }
```

What value is printed out at line 7 ?

- A) 1
- B) 2
- C) 3
- D) 0
- E) 12

Question 55:

Consider the following code:

```
1. class Question
2. {
3.     public Question(float index)
4.     {
5.     }
6.     public Question(int index)
7.     {
```

```
8.  }  
9. }
```

Which of the following are legitimate instantiating ?

- A) Question a = new Question();
- B) Question a = new Question(3,4,5);
- C) Question a = new Question(4,2);
- D) Question a = new Question(8);
- E) Question a = new Question("sss");

Question 56:

Given the code below:

```
31.     class Zombit  
32.     {  
33.         static int bomba;  
34.         public static void main(String args[])  
35.         {  
36.             Zombit zombit1 = new Zombit(12);  
37.             Zombit zombit2 = new Zombit(13);  
38.             Zombit zombit3 = new Zombit(12);  
39.             System.out.println("bomba equals to : " + bomba);  
40.         }  
41.         Zombit(int number)  
42.         {  
43.             bomba += number;  
44.         }  
45.     }
```

The output is:

- () A) bomba equals to 35
- () B) bomba equals to 33
- () C) bomba equals to 30
- () D) bomba equals to 37

Question 57:

Given the code below:

```
37. class Plomba
38. {
39.     int id;
40.     Plomba()
41.     {
42.         id = 8;
43.     }
44.     Plomba(int id)
45.     {
46.         this.id = id;
47.     }
48. }

28. class App
29. {
30.     public static void main(String args[])
31.     {
32.         Plomba plomby1 = new Plomba();
33.         Plomba plomby2 = new Plomba(72);
34.         System.out.println((plomby1.id + plomby2.id));
35.     }
```


36. }

The output is:

- A) 080
- B) 80
- C) 0
- D) 800
- E) The code doesn't compile successfully

Question 58:

Consider the following application:

```
1.class Question66
2.{
3.    public static void main(String args[])
4.    {
5.        Zombit z = new Zombit();
6.        int number = 99;
7.        z.jump(number);
8.        System.out.println(z.getValue());
9.    }
10. }
```

```
11. public class Zombit
12. {
13.     public int value = 8;
14.     public int getValue()
15.     {
16.         return value;
```

```
17. }
18. public void jump(int val)
19. {
20.     val++;
21. }
22. }
```

What value is printed out at line 7 ?

- A) 80
- B) 88
- C) 99
- D) 98
- E) 8

Question 59:

Consider the following code:

```
1. class Question
2. {
3.     public Question(double index)
4.     {
5.     }
6.     public Question()
7.     {
8.     }
9. }
```

Which of the following are legitimate instantiating ?

- A) Question a = new Question();
- B) Question a = new Question(3,2,1);

- C) Question a = new Question(4,2);
- D) Question a = new Question('a');
- E) Question a = new Question("sss");

Question 60:

Given the code below:

```
1. class Zombit
2. {
3.     static int bomba=1;
4.     public static void main(String args[])
5.     {
6.         Zombit zombit1 = new Zombit(2);
7.         Zombit zombit2 = new Zombit(3);
8.         Zombit zombit3 = new Zombit(2);
9.         System.out.println("bomba equals to : " + bomba);
10.    }
11.    private Zombit(int number)
12.    {
13.        bomba *= number;
14.    }
15. }
```

The output is:

- A) bomba equals to 15
- B) bomba equals to 16
- C) bomba equals to 7
- D) bomba equals to 0
- E) bomba equals to 12

Question 61:

Given the code below:

```
1. class Zombit extends Banga
2. {
3.     public Zombit(int j)
4.     {
5.     }
6.     public Zombit(int i, int j)
7.     {
8.         super(i,j);
9.     }
10. }
```

Which of the following are legitimate calls to instantiate instances of the Zombit class ?

- A) Zombit zom = new Zombit();
- B) Zombit zom = new Zombit("gugu");
- C) Zombit zom = new Zombit(1,2,3);
- D) Zombit zom = new Zombit(2,3);
- E) Zombit zom = new Zombit(9);

Question 62:

Given the code below,

```
1. public class Try
2. {
3.     private static int sum = 20;
4.     public static void main(String args[])
```

```
5.  {
6.      Try t1 = new Try();
7.      Try.sum++;
8.      Try t2 = new Try();
9.      t2.sum++;
10.     t1=new Try();
11.     t1.sum--;
12.     Try.sum--;
13.     System.out.println(--sum);
14. }
15.}
```

- A) 19
- B) 22
- C) 20
- D) The program doesn't compile
- E) 21

Question 63:

Given the code below:

```
1.  class Plumba
2.  {
3.      private int id = 9;
4.      private Plumba()
5.      {
6.
7.      }
8.      private Plumba(int id)
9.      {
```

```
10.         this.id = id;
11.     }
12. }

13. class App
14. {
15.     public static void main(String args[])
16.     {
17.         Plumba plomby1 = new Plumba();
18.         Plumba plomby2 = new Plumba(72);
19.         System.out.println("" + (plomby1.id + plomby2.id));
20.     }
21. }
```

The output is:

- A) 072
- B) 72
- C) 0
- D) 81
- E) The code doesn't compile successfully

Question 64:

Given the following class definition:

```
1. public class GreatPoint extends Point
2. {
3.     public GreatPoint(int num)
4.     {
5.     }
6.     public GreatPoint(int num1, int num2, int num3)
7.     {
```

```
8.         super(num1,num2,num3);
9.     }
10.}
```

Which of the following is a legitimate instantiating of the class GreatPoint ?

- A) GreatPoint gp = new GreatPoint();
- B) GreatPoint gp = new GreatPoint(4.5);
- C) GreatPoint gp = new GreatPoint(3,2);
- D) GreatPoint gp = new GreatPoint((short)(1.0),3,4);
- E) GreatPoint gp = new GreatPoint(3);

Question 65:

Which of the following sentences are true?

- A) Constructor has the same name the class has.
- B) Like other methods, the constructor must have a returned value.
- C) Every class has at least one constructor.
- D) The default constructor (the one that exists automatically in every class) is canceled when you declare a new one.
- E) The constructor can't be private.

Question 66:

Given the following code:

```
1. class Test
2. {
3.     private int num;
4.     public static void main(String args[])
5.     {
6.         Test test = new Test();
7.         test.num = 100;
8.         test.jump(test);
9.         System.out.println(test.num);
10.    }
11.    private void jump(Test other)
12.    {
13.        num+=other.num;
14.    }
15.}
```

The output is:

- A) 100
- B) 200
- C) The code doesn't compile
- D) 0
- E) 400

Question 67:

Which of the following access modifiers a normal class (not an inner class) can have?

- A) private
- B) package friendly (default)
- C) protected
- D) public

Question 68:

Which of the following access modifiers a constructor can have?

- A) private
- B) package friendly (default)
- C) protected
- D) public

Question 69:

Given the following code:

```
1. public class Demo
2. {
3.     int magicNum;
4.     public void Demo()
5.     {
6.         magicNum = 97;
7.     }
```

```
8.     public int getVal()
9.     {
10.         return magicNum;
11.     }
12.     public static void main(String args[])
13.     {
14.         Demo demo = new Demo();
15.         System.out.println(demo.getVal());
16.     }
17. }
```

The output is:

- A) 0
- B) 97
- C) 1
- D) 102

Question 70:

Given the following code:

```
1.     public class Demo
2.     {
3.         int number = 7;
4.
5.         public static void main(String args[])
6.         {
7.             int number = 8;
8.             Demo demo = new Demo();
9.             System.out.println(demo.number);
10.        }
11. }
```

The output is:

- A) 7
- B) 8
- C) The code won't compile
- D) 15
- E) 0

Question 71:

Given the following code:

```
12. public class Demo
13. {
14.     int number = 7;
15.
16.     int getNumber()
17.     {
18.         int number = 15;
19.         return number;
20.     }
21.     public static void main(String args[])
22.     {
23.         int number = 8;
24.         Demo demo = new Demo();
25.         System.out.println(demo.getNumber());
26.     }
27. }
```

The output is:

- A) 7
- B) 8
- C) The code won't compile
- D) 15
- E) 0

Question 72:

Given the following code,

```
class SQRTDemo
{
    public static void main(String args[])
    {
        Math m;
        double num=9, result;
        result = m.sqrt(num);
        System.out.println(result);
    }
}
```

The output is:

- A) Runtime Exception will be thrown
- B) 3
- C) The code doesn't compile
- D) 15

E) 0

Question 73:

True or False:

The Java programming language only passes arguments by value. When a method, that its parameter is a class type parameter, is invoked, the argument it gets is a "reference" of a specific object. The contents of the object that its reference was sent to the method can be changed but the object reference can never be changed.

A) True

B) False

Question 74:

True or False:

The static block code executes only once, when the class is loaded. The static block is usually used to initialize static variables.

A) True

B) False

Question 75:

True or False:

A final variable can be set only once. The assignment can occur independently of the declaration. If the final variable isn't set in its declaration then it must be set in every constructor.

- A) True
- B) False

Question 76:

True or False:

Calling a constructor using "new" (new Xxx() for instance) results, among other things, in the following:

1. Space for the new object is allocated and the instance variables are initialized to their default values (e.g. 0, null,false ...).
2. Explicit attribute initialization is performed.
3. The constructor that was called is executed.
4. The reference of the new object is returned.

- A) True
- B) False

Question 77:

True or False:

The JVM requires the classes to implement the `java.lang.Cloneable` interface to explicitly declare an object as having the capability to be cloned. Once an object implements the `Cloneable` interface invoking the `clone()` method (the one that was declared in `Object`) on it returns a shallow copy. If the object's class doesn't implement the `Cloneable` interface then a `CloneNotSupportedException` will be thrown.

- A) True
- B) False

Question 78:

True or False:

The shallow copy of an object that the `clone()` method (the one that was declared in the `Object` class) returns is a copy in which only the primitive and reference values are copied. This means that the existing object and the new one might share objects' references.

- A) True
- B) False

Question 79:

Given the following code:

```
1.  public class CarDemo
2.  {
3.      public static void main(String args[])
4.      {
5.          Car car1, car2;
6.          car1 = new Car("TOYOTA", 2000, 180, new Owner("David",123123));
7.          car2 = (Car)car1.clone();
8.          car2.setOwnerName("Moshe");
9.          car2.setBrand("HONDA");
10.         car2.setEngineVolume(3200);
11.         car2.setMaxSpeed(280);
12.         System.out.println(car1);
13.         System.out.println(car2);
14.     }
15. }

16. class Owner
17. {
18.     private String name;
19.     private long id;
20.     Owner(String name, long id)
21.     {
22.         this.name = name;
23.         this.id = id;
24.     }
25.     public void setName(String str)
26.     {
27.         name = str;
```



```
28.     }  
  
29.     public void setId(long val)  
30.     {  
31.         id = val;  
32.     }  
33.     public String getName()  
34.     {  
35.         return name;  
36.     }  
37.     public String toString()  
38.     {  
39.         return name+", "+id;  
40.     }  
41. }
```

```
42. class Car implements Cloneable  
43. {  
44.     private String brand;  
45.     private double engineVolume;  
46.     private int maxSpeed;  
47.     private Owner owner;  
48.     Car(String brand, double engineVolume, int maxSpeed, Owner owner)  
49.     {  
50.         this.brand = brand;  
51.         this.engineVolume = engineVolume;  
52.         this.maxSpeed = maxSpeed;  
53.         this.owner = owner;  
54.     }  
55.     public void setBrand(String str)  
56.     {  
57.         brand = str;
```

```
58.     }
59.     public void setEngineVolume(double val)
60.     {
61.         engineVolume = val;
62.     }
63.     public void setMaxSpeed(int val)
64.     {
65.         maxSpeed = val;
66.     }
67.     public void setOwner(Owner other)
68.     {
69.         owner = other;
70.     }
71.     public String getOwnerName()
72.     {
73.         return owner.getName();
74.     }
75.     public void setOwnerName(String name)
76.     {
77.         owner.setName(name);
78.     }
79.     public String toString()
80.     {
81.         return brand+", "+engineVolume+", "+maxSpeed+", "+owner;
82.     }
83.     public Object clone()
84.     {
85.         try
86.         {
87.             return super.clone();
88.         }
89.         catch(CloneNotSupportedException e)
90.         {
```

```
91.         throw new InternalError();
92.     }
93. }
94. }
```

The output is:

- () A) TOYOTA, 2000.0, 180, Moshe, 123123
HONDA, 3200.0, 280, Moshe, 123123
- () B) TOYOTA, 2000.0, 180, Moshe, 123123
TOYOTA, 2000.0, 180, Moshe, 123123
- () C) HONDA, 2000.0, 180, Moshe, 123123
HONDE, 2000.0, 180, Moshe, 123123
- () D) TOYOTA, 2000.0, 180, David, 123123
TOYOTA, 2000.0, 180, Moshe, 123123
- () E) None of the answers is true.