一：单项选择题(共 15 题，每题 2 分)

1. 下列语句哪一个正确( )

A. Java 程序经编译后会产生 machine code

B. Java 程序经编译后会产生 byte code

C. Java 程序经编译后会产生 DLL

D. 以上都不正确

答案：B

2.下列说法正确的有( )

A. class 中的 constructor 不可省略

B. constructor 必须与 class 同名，但方法不能与 class 同名

C. constructor 在一个对象被 new 时执行

D. 一个 class 只能定义一个 constructor

答案：C

3.下列哪一种叙述是正确的( )

A. abstract 修饰符可修饰字段、方法和类

B. 抽象方法的 body 部分必须用一对大括号{ }包住

C. 声明抽象方法，大括号可有可无

D. 声明抽象方法不可写出大括号

答案：D

4.下列语句正确的是( )

A. 形式参数可被视为 local variable

B. 形式参数可被字段修饰符修饰

C. 形式参数为方法被调用时，真正被传递的参数

D. 形式参数不可以是对象

答案：A

5.以下代码运行输出是（）

```
public class Person{
private String name=”Person”;
int age=0;
}
public class Child extends Person{
public String grade;
public static void main(String[] args){
Person p = new Child();
System.out.println(p.name);
}
}
```

A) 输出：Person
B) 没有输出
C) 编译出错
D) 运行出错

答案：C

6.下列哪种说法是正确的( )

A. 实例方法可直接调用超类的实例方法

B. 实例方法可直接调用超类的类方法

C. 实例方法可直接调用其他类的实例方法

D. 实例方法可直接调用本类的类方法

答案：D

7. 以下程序的运行结果是（）

```
class Person{
public Person(){
System.out.println(“this is a Person”);
}
}
public class Teacher extends Person{
private String name=”tom”;
```

```
public Teacher(){
System.out.println(“this is a teacher”);
super();
}
public static void main(String[] args){
Teacher teacher = new Teacher();
System.out.println(this.name);
}
}
```

A) this is a Person
this is a teacher
tom
B) this is a teacher
this is a Person
tom
C) 运行出错
D) 编译有两处错误

答 案： D

8.访问修饰符作用范围由大到小是（）
A) private-default-protected-public
B) public-default-protected-private
C) private-protected-default-public
D) public-protected-default-private

答 案： D

9.以下（）不是 Object 类的方法
A) clone（）
B) finalize（）
C) toString（）
D) hasNext（）

答 案： D

10．以下（）添加到 ComputerBook 中不会出错

```
class Book{
protected int getPrice(){
return 30;
}
}
public class ComputerBook extends Book{
```

}
A) protected float getPrice(){}
B) protected int getPrice(int page){}
C) int getPrice(){}
D) public int getPrice(){return 10;}

答案：D

11.以下代码，描述正确的有（）
interface IDemo{
public static final String name;1
void print();2
public void getInfo();3
}
abstract class Person implements IDemo{4
public void print(){
}
}
A) 第 1 行错误，没有给变量赋值
B) 第 2 行错误，方法没有修饰符
C) 第 4 行错误，没有实现接口的全部方法
D) 第 3 行错误，没有方法的实现

答案：A

12.以下程序运行结果是（）
public class Test extends Father{
private String name=”test”;
public static void main(String[] args){
Test test = new Test();
System.out.println(test.getName());
}
}
class Father{
private String name=”father”;
public String getName() {
return name;
}
}
A) father
B) test

C) 编译出错
D) 运行出错，无输出

答案：A

13.以下对异常的描述不正确的有（）
A) 异常分为 Error 和 Exception
B) Throwable 是所有异常类的父类
C) Exception 是所有异常类父类
D) Exception 包括 RuntimeException 和 RuntimeException 之外的异常

答案：C

14.下面代码运行结果是（B）

```
public class Demo{
public int add(int a,int b){
try{
return a+b;
}catch(Exception e){
System.out.println(“catch 语句块”);
}finally{
System.out.println(“finally 语句块”);
}
return 0;
}
public static void main(String[] args){
Demo demo = new Demo();
System.out.println(“和是：”+demo.add(9,34));
}
}
```

A) 编译异常
B) finally 语句块 和是：43
C) 和是：43 finally 语句块
D) catch 语句块 和是：43

答案：B

15.在 Java 中，关于 HashMap 类的描述，以下错误的是(　)。
A) HashMap 使用键/值得形式保存数据
B) HashMap 能够保证其中元素的顺序
C) HashMap 允许将 null 用作键
D) HashMap 允许将 null 用作值

答案：B

二：不定项选择题(共 10 题，每题 3 分)

1.下列说法正确的有( )

A. 环境变量可在编译 source code 时指定

B. 在编译程序时，所能指定的环境变量不包括 class path

C. javac 一次可同时编译数个 Java 源文件

D. javac.exe 能指定编译结果要置于哪个目录(directory)

答案：BCD

2.下列说法错误的有( )

A. 数组是一种对象

B. 数组属于一种原生类

C. int number=[]={31,23,33,43,35,63}

D. 数组的大小可以任意改变

答案：BCD

3.不能用来修饰 interface 的有( )

A.private

B.public

C.protected

D.static

答案：ACD

4. 以下属于面向对象的特征的是（）

A) 重载

B) 重写

C) 封装

D) 继承

答案： C,D

5.以下关于 final 关键字说法错误的是（）

A) final 是 java 中的修饰符，可以修饰类、接口、抽象类、方法和属性

B) final 修饰的类肯定不能被继承

C) final 修饰的方法不能被重载

D) final 修饰的变量不允许被再次赋值

答案： A,C

6.根据下面的代码，

String s = null;

会抛出 NullPointerException 异常的有（）。

A) if((s!=null) & (s.length()>0))

B) if((s!=null) & & (s.length()>0))

C) if((s==null) | (s.length()==0))

D) if((s==null) || (s.length()==0))

答案： A,C

7.以下关于对象序列化描述正确的是（）

A) 使用 FileOutputStream 可以将对象进行传输

B) 使用 PrintWriter 可以将对象进行传输

C) 使用 ObjectOutputStream 类完成对象存储，使用 ObjectInputStream 类完成对象读取

D) 对象序列化的所属类需要实现 Serializable 接口

答案： C,D

8. 在 Java 中，( )类可用于创建链表数据结构的对象。

A) LinkedList

B) ArrayList

C) Collection

D) HashMap

答案： A

9.以下对 TCP 和 UDP 描述正确的是（）

A) TCP 不能提供数据的可靠性

B) UDP 能够保证数据库的可靠性

C) TCP 数据传输效率高于 UDP

D) UDP 数据传输效率高于 TCP

答案：D

10.Java 中的集合类包括 ArrayList、LinkedList、HashMap 等类，下列关于集合类描述错误的是（）

A) ArrayList 和 LinkedList 均实现了 List 接口

B) ArrayList 的访问速度比 LinkedList 快

C) 添加和删除元素时，ArrayList 的表现更佳

D) HashMap 实现 Map 接口，它允许任何类型的键和值对象，并允许将 null 用作键或值

答案：C

三:简答题(共 4 题，每题 5 分)

1.请讲述 String 和 StringBuffer 的区别。

答：String 类所定义的对象是用于存放“长度固定”的字符串。

StringBuffer 类所定义的对象是用于存放“长度可变动”的字符串。

2. 谈谈你对抽象类和接口的理解。

答：定义抽象类的目的是提供可由其子类共享的一般形式、子类可以根据自身需要扩展抽象类、抽象类不能实例化、抽象方法没有函数体、抽象方法必须在子类中给出具体实现。他使用 extends 来继承。

接口：一个接口允许一个类从几个接口继承而来，Java 程序一次只能继承一个类但可以实现几个接口，接口不能有任何具体的方法，接口也可用来定义可由类使用的一组常量。其实现方式是 interface 来实现。

3、Java 多线程，分析 sleep()和 wait()方法的区别。

答： Sleeping 睡眠的意思 : sleep() 方法用来暂时中止执行的线程。在睡眠后，线程将进入就绪状态。

waiting 等待的意思: 如果调用了 wait() 方法，线程将处于等待状态。用于在两个或多个线程并发运行时。

4．请说出你所知道的线程同步的方法

wait():使一个线程处于等待状态，并且释放所持有的对象的 lock。
sleep():使一个正在运行的线程处于睡眠状态，是一个静态方法，调用此方法要捕捉 InterruptedException 异常。
notify():唤醒一个处于等待状态的线程，注意的是在调用此方法的时候，并不能确切的唤醒某一个等待状态的线程，而是由 JVM 确定唤醒哪个线程，而且不是按优先级。
Allnotity():唤醒所有处入等待状态的线程，注意并不是给所有唤醒线程一个对象的锁，而是让它们竞争。

四:设计题(共 4 题，每题 5 分)

1. 从键盘读入 10 个整数，然后从大到小输出。（笔试）

```
package test;

import java.util.Arrays;
import java.util.Comparator;
import java.util.Scanner;

public class Test {

public static void main(String[] args) {
Scanner in = new Scanner(System.in);
// 注意这里的数组，不是 int 的
Integer[] arr = new Integer[10];
for (int i = 0; i < 10; i++) {
arr = in.nextInt();
}
Arrays.sort(arr, new Comparator() {
@Override
public int compare(Integer o1, Integer o2) {
if (o1 > o2) return -1;
if (o1 < o2) return 1;
return 0;
}

});
System.err.println(Arrays.toString(arr));
}

}
```

2. 仔细阅读下面的程序代码，写出程序运行的输出结果。

```
class Test1
{
   private int i = 1;
   public class Test11{
        private int i = 2;
        public void methodI(int i)
        {
             i++;
             this.i++;
              Test1.this.i++;
             System.out.println("i of methodI():"+i);
             System.out.println("i of Test11:"+this.i);
             System.out.println("i of Test1:"+Test1.this.i);
        }
   }
   Test11 ic=new Test11();
   public void increaseI(int k)
   {
        ic.methodI(k);
   }
```

```
    public static void main(String [] args)
    {
        Test1 oc=new Test1();
        oc.increaseI(20);
    }
}
```

答案：

```
i of methodI():21
i of Test11:3
i of Test1:2
```

3. 仔细阅读下面的程序代码，若经编译和运行后，请写出打印结果。

```
public class Test
{
       public static void main(String args[])
       {
    int [ ] a = {10, 20, 30, 40, 50};
        int s =0;
    for (int c: a)
            s +=c;
    System.out.print(s );
    }
}
```

打印结果：150

4.仔细阅读下面的程序代码，若经编译和运行后，请写出打印结果。

```
class myException extends Exception{}
public class Sample{
    public void foo(){
        try{
            System.out.print(1);
            bar();
            System.out.print(2);
        }catch(myException e){
            System.out.print(3);
            }
        finally{
            System.out.print(4);
        }
    }
    public void bar() throws myException{
        throw new myException();
    }
    public static void main(String args[]){
        Sample s=new Sample();
        s.foo();
```

```
    }
}
```

打印结果：134

5、Java 的通信编程，编程题(或问答)，用 JAVA SOCKET 编程，读服务器几个字符，再写入本地显示？

答:Server 端程序:

```
package test;

import java.net.*;

import java.io.*;

public class Server

{

private ServerSocket ss;

private Socket socket;

private BufferedReader in;

private PrintWriter out;

public Server()

{

try

{

ss=new ServerSocket(10000);

while(true)

{

socket = ss.accept();
```

```
String RemoteIP = socket.getInetAddress().getHostAddress();

String RemotePort = ":"+socket.getLocalPort();

System.out.println("A client come in!IP:"+RemoteIP+RemotePort);

in = new BufferedReader(new


InputStreamReader(socket.getInputStream()));

String line = in.readLine();

System.out.println("Cleint send is :" + line);

out = new PrintWriter(socket.getOutputStream(),true);

out.println("Your Message Received!");

out.close();

in.close();

socket.close();

}

}catch (IOException e)

{

out.println("wrong");

}

}

public static void main(String[] args)

{

new Server();
```

```
}

};
```

Client 端程序:

```
package test;

import java.io.*;

import java.net.*;


public class Client

{

Socket socket;

BufferedReader in;

PrintWriter out;

public Client()

{

try

{

System.out.println("Try to Connect to 127.0.0.1:10000");

socket = new Socket("127.0.0.1",10000);

System.out.println("The Server Connected!");

System.out.println("Please enter some Character:");

BufferedReader line = new BufferedReader(new
```

```
InputStreamReader(System.in));

out = new PrintWriter(socket.getOutputStream(),true);

out.println(line.readLine());

in = new BufferedReader(new InputStreamReader(socket.getInputStream()));

System.out.println(in.readLine());

out.close();

in.close();

socket.close();

}catch(IOException e)

{

out.println("Wrong");

}

}

public static void main(String[] args)

{

new Client();

}

};
```