

Bob Technologies Placement Paper – 2

Questions

1)What will happen when you attempt to compile and run this code?

```
abstract class Base{
abstract public void myfunc(); public
void another(){
System.out.println("Another method");
}
}
```

```
public class Abs extends Base{
public static void main(String argv[]){
Abs a = new Abs();
a.amethod();
}
public void myfunc(){
System.out.println("My Func");
}
public void amethod(){
myfunc();
}
}
```

- 1) The code will compile and run, printing out the words "My Func"
- 2) The compiler will complain that the Base class has non abstract methods
- 3) The code will compile but complain at run time that the Base class has non abstract methods
- 4) The compiler will complain that the method myfunc in the base class has no body, nobody at all to looove it

2)What will happen when you attempt to compile and run this code?

```
public class MyMain{
public static void main(String argv){
System.out.println(" Hello cruel world");
}
}
```

- 1) The compiler will complain that main is a reserved word and cannot be used for a class
- 2) The code will compile and when run will print out "Hello cruel world"
- 3) The code will compile but will complain at run time that no constructor is defined
- 4) The code will compile but will complain at run time that main is not correctly defined

3)Which of the following are Java modifiers?

- 1) public 2) private 3) friendly 4) transient 5) vagrant

4)What will happen when you attempt to compile and run this code?

```
class Base{
abstract public void myfunc(); public
void another(){
System.out.println("Another method");
}
}
```

```
public class Abs extends Base{
public static void main(String argv[]){
Abs a = new Abs();
a.amethod();
}
```

```
public void myfunc(){
System.out.println("My func");
}
```

```
public void amethod(){
myfunc();
}
}
```

- 1) The code will compile and run, printing out the words "My Func"
- 2) The compiler will complain that the Base class is not declared as abstract .
- 3) The code will compile but complain at run time that the Base class has non abstract methods
- 4) The compiler will complain that the method myfunc in the base class has no body , nobody at all to loove it

5)Why might you define a method as native?

- 1) To get to access hardware that Java does not know about
- 2) To define a new data type such as an unsigned integer
- 3) To write optimized code for performance in a language such as C/C++
- 4) To overcome the limitation of the private scope of a method

6)What will happen when you attempt to compile and run this code?

```
class Base{
public final void amethod(){
System.out.println("amethod");
}
}
```

```
public class Fin extends Base{
public static void main(String argv[]){
Base b = new Base();
b.amethod();
}
}
```

- 1) Compile time error indicating that a class with any final methods must be declared final itself

- 2) Compile time error indicating that you cannot inherit from a class with final methods
- 3) Run time error indicating that Base is not defined as final
- 4) Success in compilation and output of "amethod" at run time.

7)What will happen when you attempt to compile and run this code?

```
public class Mod{
public static void main(String argv[]){
}
public static native void amethod();
}

```

- 1) Error at compilation: native method cannot be static
- 2) Error at compilation native method must return value
- 3) Compilation but error at run time unless you have made code containing native amethod available
- 4) Compilation and execution without error

8)What will happen when you attempt to compile and run this code?

```
private class Base{}
public class Vis{
transient int iVal;
public static void main(String elephant[]){
}
}

```

- 1)Compile time error: Base cannot be private
- 2)Compile time error indicating that an integer cannot be transient
- 3)Compile time error transient not a data type
- 4)Compile time error malformed main method

9)What happens when you attempt to compile and run these two files in the same directory ?

```
//File P1.java
package MyPackage;
class P1{
void afancymethod(){
System.out.println("What a fancy method");
}
}

```

```
//File P2.java
public class P2 extends P1{
public static void main(String argv[]){
P2 p2 = new P2();
p2.afancymethod();
}
}

```

- 1) Both compile and P2 outputs "What a fancy method" when run
- 2) Neither will compile
- 3) Both compile but P2 has an error at run time
- 4) P1 compiles cleanly but P2 has an error at compile time

Question 10)

You want to find out the value of the last element of an array. You write the following code. What will happen when you compile and run it.?

```
public class MyAr{
public static void main(String argv[]){
int[] i = new int[5];
System.out.println(i[5]);
}
}
```

- 1) An error at compile time
- 2) An error at run time
- 3) The value 0 will be output
- 4) The string "null" will be output

11) You want to loop through an array and stop when you come to the last element. Being a good java programmer and forgetting everything you ever knew about C/C++ you know that arrays contain information about their size. Which of the following can you use?

1) myarray.length(); 2) myarray.length; 3) myarray.size 4) myarray.size(),

12) What best describes the appearance of an application with the following code?

```
import java.awt.*;
public class FlowAp extends Frame{
public static void main(String argv[]){
FlowAp fa=new FlowAp();
fa.setSize(400,300);
fa.setVisible(true);
}
```

```
FlowAp(){
add(new Button("One"));
add(new Button("Two"));
add(new Button("Three"));
add(new Button("Four"));
} //End of constructor
```

//End of Application

- 1) A Frame with buttons marked One to Four placed on each edge.
- 2) A Frame with buttons marked One to four running from the top to bottom
- 3) A Frame with one large button marked Four in the Centre
- 4) An Error at run time indicating you have not set a LayoutManager

13) How do you indicate where a component will be positioned using FlowLayout?

- 1) North, South, East, West
- 2) Assign a row/column grid reference
- 3) Pass a X/Y percentage parameter to the add method
- 4) Do nothing, the FlowLayout will position the component

14)How do you change the current layout manager for a container

- 1) Use the setLayout method
- 2) Once created you cannot change the current layout manager of a component
- 3) Use the setLayoutM anager method
- 4) Use the updateLayout method

15)Which of the following are fields of the GridBagConstrai nts class ?

- 1) ipadx 2) fill 3) insets 4) width

16)What most closely matches the appearance when this code runs ?

```
import java.awt.*;  
public class CompLay extends Frame {  
    public static void main(String argv[]){  
        CompLay cl = new CompLay();  
    }  
}
```

```
CompLay(){  
    Panel p = new Panel();  
    p.setBackground(Color. pink);  
    p.add(new Button("One"));  
}
```

100Duniya