Java Assignment 1-B (Theory)

Q-1 Is it possible to override or overwrite a static method in java?

-> ~~Yes, it is possible to overloa~~

-> No, it is not possible to override a static method because method overriding is based upon dynamic binding at runtime and static methods are bonded using static binding at compile time

Q-2 How can you execute a code before the main method?

-> Yes, we can execute a code before main method by using static blocks

-> Static blocks are executed before main method at the time of class loading.

=>
```
class Test
  {
    static
       { System.out.println("Hi"); }
    public static void main(String [] args)
       { System.out.println("Hi Main"); }
  }
```

* Output:-

->
```
Hi
Hi Main
```

Q-3 Can you use default constructor of class even if explicit constructor is defined?

→ No, you cannot use default constructer if you have defined explicit constructor.

→ It is because, if you define any constructor, then it cannot be a default constructor, even if we write the constructor exactly same as default constructor then it falls in category of No-Argument user defined constructor.

→ Also, default constructor is generated by compiler if we donot create an explicit constructor but if we create one then default constructor is not generated.

Q-4 Explain the use of super and this keyword.

→ * Super:-

→ "super" is a reserved keyword in java

→ super keyword is used to refer Super class's instance members and static members.

* this:-

→ "this" is also a reserved keyword in java

→ this keyword is used to reffer current class's instance membeos and Static members.

e.g

```
class Parent
  { int x=10;
   Static int y= 120;
  }
class child extends Parent
  {   void print()
       { System.out.println(this.x);
         System.out.println(this.y);
       }
     int x= 15;
  Static int y= 125;
        void printSuper()
    { system.out.println(super.x);
     System.out.println(super.y);
    }
  ~~}~~ public static void man(String[] args)
     {
       chid c1= new child();
        c1.print();
       c1.printSuper();
      }
  }
```

* output :-

```
15
125
10
120
```

Q-5 If you have multiple constructors in class, can you call constructor from another constructor?

→ yes, we have multiple constructors in class and we can call constructor from another constructor.

→
```
class Test
  {  Test()
     { System.out.println("test-1");
     }
     Test(int x)
      { Test();
        System.out.println("test-2");
      }
   public static void main (String[] arg)
     {
        Test t=new Test(10);
     }
  }
```

* Output:-

```
test-1
test-2
```

Q-6 Can you use different return types of methods when it is overriden?

→ Before JDK 5, it is not possible to override method by changing the return type because when we override method in class then it has to be exactly same as that of parent class method.

-> From java5.0 onwards it is possible to have different return type for a overriding method in child class, but child's returntype should be sub-type of parent's return type

Q-7 what will be order of call of constructors in inheritance?

-> order of call of constructors in inheritance is defined in java.

-> First Parent class/ Superclass constructor called and then the subsequent child class constructors. is ~~called~~

=>
```
class Parent
      {
        Parent()
        { System.out.println("Parent"); }
class child extends Parent
      {
        Child()
        { System.out.println("Child"); }
      }
public class Test
    { ~~System.ou~~
    public static void main(String[] args)
      { child c1 = new Child(); }
    }
```

* Output:

->
```
Parent
Child
```

Q-8 What if in program is written as static public void instead of public static void?

→ Yes, we can change the order of public static void main() to static public void main()
→ Here, java compiler doesn't throw any compile time or runtime error.
→ But it is recomended to write public static void.

Q-9 Why is main method static?

→ main() method is static because compiler can call it without the creation of an object or before the creation of object
→ main() method is starting point of execution
→ Static method of class can be called by using the class name only without creating an object of class
→ If main() is allowed to be non-static, then while calling main() methods JVM has to instantiate the class and it will be ambiguity error

Q-10 Explain JRE, JDK, JVM.
→ JVM :-

→ JVM stands for JAVA VIRTUAL MACHINE
→ It is an abstract machine that enables your computer to run java program
→ JVM translates byte code into native machine code
→ JVM is platform dependent

→ JRE :-

→ JRE stands for JAVA RUNTIME ENVIRONMENT

→ It is a software package that provides Java class liberies , JVM and other components required to run java application

→ JDK :-

→ JDK stands for Java Development kit

→ It is a software development kit required to develop applications in Java.