Assignment No. 6

DATE / / 20

Q.1 What is method overloading in Java & explain with an example?

→ method overloading → having two or more methods in class with same name & different arguments (or parameters).

↳ It can be with a different number of arguments or different data types of arguments.

eg: 1)

~~class Addition {~~

~~static int add (int a, int b) {~~

~~return~~

```
class Addition {
public int add (int x, int y) {
  return (x + y); }

public int add ( int x, int y, int z){
return (x + y + z); }

public double add (double x , double y){
return (x +y ); }

public static void main (string [ ] args) {
 sum a= new Addition( );

 System.out.println (a. add (10,20));
 System.out.println (a. add (10,20,30));
 System.out.println (a. add (10.5, 20.5));
 }
}
```

Output: 30 60 31.0.

PAGE NO.:
DATE.: / / 20

Q.2 What are the rules for method overloading resolution in Java? How does java determine which overload method to call?

→ Rules of method overloading :-

1. The overloaded and overloading methods must be in same class
2. method parameters must change: either the number or the type of parameters must be different in the two methods.
3. The return type can be freely modified.
4. The access modifier (public, private, & so on) can be freely modified.
5. thrown exceptions, if any, can be freely modified.

→ method overloading is determined at compile time. Hence, it is also known as compile time polymorphism.

DATE: / / 20

Q3 What does the static keyword mean in java? Explain diff. b/w static & non-static methods.

→ Static keyword in java indicates that a particular member is not a ~~p~~ instance, but rather ~~than~~ part of a type.

↳ If any member in a class is declared as static, it means that even before the class is initiated, all the static member can be accessed & become active.

~~kep~~

Static	Non-static
1. static method that belongs to class but it does not belong to instance of that class & this method can called without instance or obj of class	1. Every method in java defau-lt to non-static method without a static keyword preceding it. non-static method & static variable also without using obj class.
2. ~~an~~ method can only access only static data member & static methods of another class or same class but cannot access non-static method & variable.	2. method can access static data member & static data methods as well as non-static members & methods are different class or same class.
3. Uses compile time or early binding	3. uses runtime or dyna-mic binding.
4. less memory used for exe-cution	4. Much memory used for execution.

Q.4 Can static methods be overloaded & overridden in Java? How static variable shared across multiple instances of class?

→ Static methods in Java cannotbe over~~loa~~ridden because static methods are not associated with instance of class; but with class itself.

↳ Static methods can be overload, we can have two or more static methods with same name but with different parameters.

→ Yes, To stored information that is shared across instances of a class, use a static variable. All instances of same class share a single copy of the static variable.

Q.5 What is role of static keyword in context of memory management?

→ Static keyword used for memory management.
↳ used to share the same variable or method of given class.
↳ User can apply static keywords with variables, methods, blocks & nested classes.
↳ Static keyword belong to class than an instance of class.
↳ static is used for a constant variable or method that is same for every instance of class.

Q.6. What is significance of final keyword in Java?

→ final keyword is non-access modifier used for classes, attribute & methods which makes them non-changeable (impossible to inherit or override)

↳ Useful when you want a variable to always store the same value.

↳ final keyword is called as "modifier"

final variable → to create constant variable

↳ when variable declare as final, its value cannot changed once it initialized

final methods → prevent method overriding

↳ method declare as final it cannot overriden by subclass. useful for methods that are parts of class public API & should modified by subclass.

final class → prevent inheritance.

↳ it cannot be extended by a subclass this is useful for classes that intended to be used as is and should not be modified or extended.

PAGE NO.:
DATE: / / 20

Q.7 Can a final methods be overridden in subclass?
How does the final keyword affect variables, meth-ods & classes in Java?
→ No the methods that are declared as final cannot be overridden or hidden.

↳ final keyword serves as a non- access modifier applicable to classes methods & variables.
↳ final class cannot be subclassed.
↳ non-access modifier used for classes attribute & methods which makes them non-changeable.

Q.8
H
me
→
p
↳
u
↳
of
s
→
c
↳
↳

PAGE NO.:
DATE.: / / 20

Q.8 What does this keyword represent in Java? How is this keyword used in constructors and methods?

→ 'this' keyword in java serves a fundamental purpose it refers to the current object.
↳ this represents instance of class where it's used.
↳ commonly used to access or modify the fields of the current object. when field name are same as local variable name.

→ this keyword refer to current object in method or constructor.
↳ most common use of this keyword is to elimate the confusion b/w class attributes & parameters withe same name.
↳ because class attribute is shadowed by method or constructor parameter.

PAGE NO.:
DATE.: / / 20

Q.9 . What are narrowing & widening conversions in java?
→ widening conversions :
↳ preserve source value but can change its representation . this occurs if you convert from an integral type to decimal or from char to string →(automatically) → convert small to large type
byte → short → char → int → long → float → double.
→ Narrowing conversions :
↳ it changes a value to a data type that might not be able to hold some of possible values → (manually) → convert Large to small type
double → float → Long → int → char → short → byte

Q.10. provide examples of narrowing & widening conservation b/w primitive data type.
→ eg :

```
public class main {
 public static void main(string [ ] args) {
  int myInt = 9;
  double myDouble = myInt;
 system . out . println (myInt) ;
 system . out . println (myDouble);
 }
}
```

O/P :-
```
9
9.0.
```

```
public class main{}:
public static void main(String [] args){
  double myDouble = 9.78 d;
  int myInt = (int) myDouble ;
  system. out. println (" myDouble) ;
  system. out. println (myInt);
 }
}
```

OP : 9.78
     9

Q.10: How does Java handle potential loss of precision during narrowing conversion?

→ It simply the loss of information while handling data.

↳ It corresponds to possibility of losing the value or precision of a variable while converting one type to another.

↳ when we try to assign a variable of large-sized type to smaller sized type, Java will generate an error.

↳ incompatible types: possible loss conversion, while compiling the code.

Q.12 Explain concept of automatic widening conversion in java?

→ widening conversion is automatically converting a smaller type to larger type sized.
byte → short → ~~char~~ → int → long → float → double.

↳ It changes a value to a data type that can allow for any possible value ~~of~~ original data.

↳ preserve the source value but can change its representation.

↳ It occurs if you convert from integral type to decimal, or from char to string.

Q.13. what are implication of narrowing & widening conversion on type compatibility and data loss?

```
→ ~~main~~ class widening {
      public static void main (String [] args){
        int ~~i~~ = 100;
        long l = i;
        float f = l;
        system. out. println ("Int "+ i);
        system. out. println ("Long " + l);
        system. out. println ("Float " + f);
    }
  }
```

```
o/p  Int    100
     long   100
     Float  100.0
```

eg:
```
public class narrowing {
    public static void main (String [ ] args) {
    char ch = 'c';
    int num = 88;
    ch = num;
    }
}
```

→ O/P : error will generate
because incompatible types.
i.e as integer variable takes 4 bytes while character datatype requires 2 bytes.

↳ we trying to plot data from 4 bytes to 2 byte. which is not possible.