**SRM UNIVERSITY**

**DEPARTMENT OF INFORMATION TECHNOLOGY**

**CYCLE TEST-II – Answer Key**

**IT1013- Programming in Java-Set A**

**DATE: 17.3.15 DURATION: 100 MINS**

**YEAR/SEM: II/IV TOTAL MARKS: 50**

**INSRUCTIONAL OBJECTIVES:**

To learn basic Java programming language features, new language features.

**STUDENT OUTCOMES:**

Outcome c - An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs

c2 – Implement using coding standards

Outcome i - An ability to use current techniques, skills, and tools necessary for computing practice

i1 – Ability to use current techniques and tools.

i2 – Apply skills required to solve problems

**PART-A**

**ANSWER ALL THE QUESTIONS (10\*1=10)**

1. \_\_\_\_\_\_\_\_\_\_\_\_\_\_- method cannot be overridden
2. static b. final c. abstract d. both a & b
3. class SampleDemo{

 public static void main(String args[]){

   String s="Java";

   s.concat(" Programming");

   System.out.println(s);  }   }

1. Java Programming
2. JavaProgramming
3. Java
4. Programming
5. Which two of the following are legal declarations for abstract classes and interfaces?
6. final abstract class Test {}
7. public static interface Test {}
8. final public class Test {}
9. protected abstract class Test {}
10. protected interface Test {}
11. abstract public class Test {}
12. 1 & 2 b. 2&4 c.3&6 d. 5&6

[(1) is wrong because a class cannot be *abstract* and *final*â€”there would be no way to use such a class. (2) is wrong because interfaces and classes cannot be marked as *static*. (4) and (5) are wrong because classes and interfaces cannot be marked as *protected*.]

4.Analyze the following method signature and select the statement that must be true:

private int someMethod(int a, Object b, String c, char d)

1. a is pass by reference, b is pass by value, c is pass by reference, and d is pass by value

### a is pass by value, b is pass by reference, c is pass by reference, and d is pass by value

1. a is pass by reference, b is pass by reference, c is pass by reference, and d is pass by reference
2. a is pass by reference, b is pass by value, c is pass by value, and d is pass by value
3. In a nested class which of the following is true.
4. Only outer class can be static
5. Only inner class can be static
6. Both outer and inner class can be static
7. Static is not applicable for nested class.
8. Trace the output for the following code:

class Main {

 public static void main(String args[]){

    final int i;

    i = 20;

    i = 30;

    System.out.println(i);

 }

}

a. 20 b. 30 c. compilation error d. Garbage value

7.which of the following is true?

a. A class can extend more than one class

b. An interface can implement many interface

c. A class can extend one class and implement many interface

d. A class can extend more than one class and implement more than one interface

8. Which of these is correct way of calling a constructor having no parameters, of superclass A by subclass B?.

a. super(void);.

b. superclass.();.

c. super.A();.

d. super();

9. Output of following Java Program?

|  |
| --- |
| class Base {      public void show() {         System.out.println("Base class show()is called");      }  }    class Derived extends Base {      public void show() {         System.out.println("Derived class show() is called");      }  }    public class Main {      public static void main(String[] args) {          Base b = new Derived(); //line 13          b.show();      }  } |

1. Derived class show() is called
2. Base class show() is called
3. Compilation error at line 13
4. No compilation error, during execution throws exception

10. If a variable is declared as protected , then it can be used in \_\_\_\_\_\_\_

1. Any class of any package, if the class is inherited.
2. Any class of any package
3. only in the specified class
4. only in the specified package
5. 1 only b. 2 only c. 3 only d. 1&4

**SRM UNIVERSITY**

**DEPARTMENT OF INFORMATION TECHNOLOGY**

**CYCLE TEST-II-Answer Key**

**IT1013- Programming in Java- Set B**

**DATE: 17.3.15 DURATION: 100 MINS**

**YEAR/SEM: II/IV TOTAL MARKS: 50**

**INSRUCTIONAL OBJECTIVES:**

To learn basic Java programming language features, new language features.

**STUDENT OUTCOMES:**

Outcome c - An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs

c2 – Implement using coding standards

Outcome i - An ability to use current techniques, skills, and tools necessary for computing practice

i1 – Ability to use current techniques and tools.

i2 – Apply skills required to solve problems

**PART-A**

**ANSWER ALL THE QUESTIONS (10\*1=10)**

 1. Which of these is supported by method overriding in Java?  
a) Abstraction b) Encapsulation c) Polymorphism d) inheritance

2. class output {

public static void main(String args[])

{

StringBuffer c = new StringBuffer("Hello");

c.delete(0,2);

System.out.println(c);

}

a) He  
b) Hel  
c) lo  
d) llo

3. A method without the body is called \_\_\_\_\_\_\_\_ method.

a. protected b. final c. abstract. d. user defined.

4. \_\_\_\_\_\_\_ is passed to a method by use of call-by-reference.

a. variables. b. objects. c. value d. operators.

5. What modifiers may be used with an inner class that is a member of an outer class?

a. inner class may be declared only as public, protected

b. inner class may be declared only as public, protected, private

c. inner class may be declared only as static, final, or abstract

d. inner class may be declared as public, protected, private, static, final, or abstract

6. which of these keywords is used to prevent content of a variable from being modified?  
**a) final**  b) last c) constant d) static

7. Which of the following is correct way of implementing an interface salary by class manager?  
a) class manager extends salary {}  
b) class manager implements salary {}  
c) class manager imports salary {}  
d) None of the mentioned.

8.super keyword can be used to \_\_\_\_\_.

a. call super class constructor.

b. access super class member

c. both a and b.

d. none of the above

9. What is the output of this program?

class A {

public int i;

private int j;

}

class B extends A {

void display() {

super.j = super.i + 1;

System.out.println(super.i + " " + super.j);

}

}

class inheritance {

public static void main(String args[])

{

B obj = new B();

obj.i=1;

obj.j=2;

obj.display();

}

}

a) 2 2  
b) 3 3  
c) Runtime Error  
d) Compilation Error

10. Which of the following is correct way of importing an entire package ‘pkg’?  
a) import pkg.  
b) Import pkg.  
c) import pkg.\*;  
d) Import pkg.\*;

**SRM UNIVERSITY**

**DEPARTMENT OF INFORMATION TECHNOLOGY**

**CYCLE TEST-II-Answer Key**

**IT1013- Programming in Java**

**DATE: 17.3.15 DURATION: 100 MINS**

**YEAR/SEM: II/IV TOTAL MARKS: 50**

**INSRUCTIONAL OBJECTIVES:**

To learn basic Java programming language features, new language features.

**STUDENT OUTCOMES:**

Outcome c - An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs

c2 – Implement using coding standards

Outcome i - An ability to use current techniques, skills, and tools necessary for computing practice

i1 – Ability to use current techniques and tools.

i2 – Apply skills required to solve problems

**PART- B**

**ANSWER ANY FOUR (4\*4=16)**

11. Consider the following statement.

“August 15 is celebrated as the Independence day of India”. Write a program to change 15 to 26, August to January, Independence to Republic and finally print “January 26 is celebrated as the Republic day of India”.

public class Sample {

/\*\*

\* @param args the command line arguments

\*/

public static void main(String[] args) {

// TODO code application logic here

String str = "August 15 is celebrated as the Independance day";

String word;

str=str+" ";

String newWord="\0";

char ch;

int n=0,l=str.length();

for(int i=0;i<l;i++)

{

ch=str.charAt(i);

if(ch==' ')

{

word=str.substring(n, i);

if(word.equals("August"))

// System.out.println("January");

newWord=newWord+"January ";

else if(word.equals("15"))

//System.out.println("26");

newWord=newWord+"26 ";

else if(word.equals("Independence"))

// System.out.println("Republic ");

newWord=newWord+"Republic ";

else //System.out.println(word+"");

newWord=newWord+" "+word;

n=i+1;}

}

System.out.println(newWord);

}}

1. Abstract or Interface which provide 100% abstraction. With necessary points justify your option?

Interface provide 100% abstraction because

1. Main difference is methods of a Java interface are implicitly abstract and cannot have implementations. A Java abstract class can have instance methods that implements a default behavior.
2. Variables declared in a Java interface is by default final. An  abstract class may contain non-final variables.
3. Members of a Java interface are public by default. A Java abstract class can have the usual flavors of class members like private, protected, etc..
4. Create an interface that declares methods min(), and max(). The member methods should return an int value that represents, the minimum value and the maximum value in a array.

interface MinandMax

{

int min();

int max();

}

public class NewClass implements MinandMax {

int numbers[] = new int[]{32,43,53,54,32,65,63,98,43,23};

public int min() {

int smallest = numbers[0];

for(int i=1; i< numbers.length; i++)

if(numbers[i] < smallest)

smallest = numbers[i];

return smallest;

}

public int max() {

int largest = numbers[0];

for(int i=1; i< numbers.length; i++)

if(numbers[i] > largest)

largest = numbers[i];

return largest;

}

public static void main(String args[])

{

NewClass obj=new NewClass();

System.out.println(obj.min());

System.out.println(obj.max());

}

}

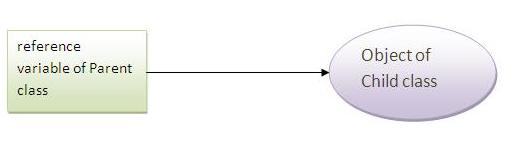
1. Explain Dynamic Method dispatch.

**Runtime polymorphism** or **Dynamic Method Dispatch** is a process in which a call to an overridden method is resolved at runtime rather than compile-time.

In this process, an overridden method is called through the reference variable of a superclass. The determination of the method to be called is based on the object being referred to by the reference variable.

### Upcasting

When reference variable of Parent class refers to the object of Child class, it is known as upcasting. For example:



**class** Bike{

**void** run(){System.out.println("running");}

}

**class** Splender **extends** Bike{

**void** run(){System.out.println("running safely with 60km");}

**public** **static** **void** main(String args[]){

    Bike b = **new** Splender();//upcasting

    b.run();

  }

}

1. List your understanding of static and final keyword.

* If you make any variable as final, you cannot change the value of final variable(It will be constant).
* Java classes declared as final cannot be extended. Restricting inheritance!
* Methods declared as final cannot be overridden.

Eg:

class Bike{

 final int speedlimit=90;//final variable

 void run(){

  speedlimit=400;  // error

 }

 public static void main(String args[]){

 Bike obj=new  Bike();

 obj.run();

 }

}

Static:

The static keyword in java is used for memory management mainly. We can apply java static keyword with variables, methods, blocks and nested class. The static variable gets memory only once in class area at the time of class loading.

Eg:

class Counter2{

static int count=0;//will get memory only once and retain its value

Counter2(){

count++;

System.out.println(count);

}

public static void main(String args[]){

Counter2 c1=new Counter2();

Counter2 c2=new Counter2();

Counter2 c3=new Counter2();

 }

}

**PART- C**

**ANSWER ALL THE QUESTIONS (2\*12=24)**

**16. a)** 1.Design a simple class called “PurchaseItem” with following variables and methods

Variables:

String name;

private double unitprice;

Methods:

public PurchaseItem( String name,double unitprice) -> to initialize the instance variables

public double getPrice()-> to returns the unitPrice.

public String toString()->to return the name of the item followed by @ symbol, then the unitPrice

2. Create two subclasses WeighedItem and CountedItem extends PurchaseItem. WeighedItem has an additional instance variable weight (double) in Kg while CountedItem has an additional variable quantity (int) both private.

private double weight // for WeightedItem class

private int quantity;// for CountedItem

3. Write an appropriate constructor for each of the classes make use of the constructor of the superclass in defining those of the subclasses.

4. Override getPrice() that returns the price of the purchasedItem based on its unit price and weight (WeighedItem), or quantity (CountedItem). Make use of getPrice of the superclass.

[hint: // in PurchaseItem class

public double getPrice(){

return unitprice;

}

// in WeightedItem class

public double getPrice(){

return super.getPrice()\*weight;

}

]

5. Override toString() for each class making use of the toString method of the superclass in defining those of the subclasses. toString() should return something that can be printed on the receipt.

[hint: // in PurchaseItem class

public String toString(){

return name+"\t@\t"+unitprice;

}

// in WeightedItem class

public String toString(){

return super.toString()+"\t"+weight+"kg\t"+getPrice()+"Rs.";

}]

6. Write a java main class where you construct objects from the two subclasses and print them as shown.

Sample output:

[banana@3.0](mailto:banana@3.0) 1.37kg 4.11Rs

[pens@4.5](mailto:pens@4.5) 10units 45.0Rs

public class PurchaseItem{

private String name;

private double unitprice;

public PurchaseItem( String name,double unitprice) {

this.unitprice = unitprice;

this.name = name;

}

public double getPrice(){

return unitprice;

}

public String toString(){

return name+"\t@\t"+unitprice;

}

}

WeightedItem.java

public class WeightedItem extends PurchaseItem

{

private double weight;

public WeightedItem(String name, double unitprice,double weight)

{

super( name,unitprice);

this.weight = weight;

}

public double getPrice(){

return super.getPrice()\*weight;

}

public String toString(){

return super.toString()+"\t"+weight+"kg\t"+getPrice()+"Rs.";

}

}

CountedItem.java

public class CountedItem extends PurchaseItem

{

private int quantity;

public CountedItem(String name, double unitprice,int quantity)

{

super(name,unitprice);

this.quantity = quantity;

}

public double getPrice(){

return super.getPrice()\*quantity;

}

public String toString(){

return super.toString()+"\t"+quantity+"units\t"+getPrice()+"Rs.";

}

}

PurchaseInfo.java

public class PurchaseInfo

{

public static void main(String[] args)

{

WeightedItem w = new WeightedItem("banana",3,1.37);

CountedItem c= new CountedItem("pens",4.5,10);

System.out.println(w);

System.out.println(c);

}

}

**(OR)**

**16. b)** Create a Vehicle class that is an abstract class defining the general details and actions associated with a vehicle. Create Car, Truck, and Minivan classes that inherit the Vehicle class. The Car, Truck, and Minivan classes should include additional member’s specific to the type of vehicle being represented. Create a main class to test the classes. [Use: minimum of 2 variables in each class. Minimum of 3 methods in each class with argument and return type.]

Two variable each carries half mark (3 Marks)

Three methods with argument and return type carries one mark (3\*3=9Marks)

[methods like

String move();

String Properties(String name, int year)

boolean isBreakApplied(int power);

int gear(int number); ……….]

**17. a.** i) Write a java program to find sum of first n natural numbers using recursion. Class sum{ (6)

int add(int n) {

if(n!=0)

return n+add(n-1);

else return 0;

}

Public static void main (String args[])

{

Sum obj = new sum ();

System.out.println("Enter an positive integer: ");

Scanner sc = new Scanner(System.in);

int n=sc.nextInt();

System.out.println(obj.add(n));

}

}

**ii**) Write a java program to add two numbers. Input the values using command line argument. (6)

class add

{

public static void main (String[] args)

{

System.out.println(Integer.parseInt(args[0]+ Integer.parseInt(args[1]);

}}

**(OR)**

**17.b** i) Justify the phrase “Strings are immutable” (3)

Immutable simply means unmodifiable or unchangeable.

Once string object is created its data or state can't be changed but a new string object is created.

Eg: (1Mark)

class SampleDemo{

 public static void main(String args[]){

   String s="Java";

   s.concat(" Programming");

   System.out.println(s);  }   }

ii) Consider the string “I am a Java Expert ”. Do the following operations

1. Extract the substring “Expert” (1)

str.substring(12);

1. Remove the extra space at the end. (1)

str.trim();

1. What is the character at position 7. (1)

Str.charAt(7)

Ans: J

**ii) W**ith supporting code explain different access control in java (6)

### Default (1 Mark)

When a method is set to default it will be accessible to the class which are defined in the same package. Any method in any class which is defined in the same package can access the given method via **Inheritance** or **Direct access**.

 Public**(1 Mark)**

When a method is set to public it will be accessible from any class available in the Java world. Any method in any class can access the given method via **Inheritance** or **Direct access** depending on class level access.

**Protected(1 Mark)**

If a method is set to protected inside a class, it will be accessible from its sub classes defined in the same or different package.

**Private(1 Mark)**

A method if defined private will be accessible only from within the class it is defined. Such methods are not accessible from outside the defined class, not even its subclass .

Eg: (2 Marks)