Q-1 Differentiate OOP and POP

Q-2: Explain basic concepts of OOP

A-2: Following are the basic concepts of OOP.

1. Object

-> Object is collection of properties and methods

-> Objects are run time entities. It can be anything on which we can create programs.

-> Object is variable of type class.

Student

Rollno, name, marks, class

Total, Result

2. Class: Class is user defined type. It bounds properties and methods into a single unit.

Object is variable of type class, so, we can create any no. of objects after defining the class.

class A

{

};

A a1, a2, a3;

3. Abstraction: Abstraction means declaration of member variables and member function without including background details. (Much more details)

4. Encapsulation: Encapsulation bounds properties and methods into a single unit.

Class is an example of encapsulation.

Class encapsulates properties and methods.

It provides Data hiding facility.

5. Inheritance: Inheritance means to acquire the properties of one class into another.

-> It provides reusability facility.

-> The class from which we inherit the properties is known as super class (Parent class) and new class is called sub class (Child class).

6. Polymorphism: Ability to take more than one form.

+ operator, if we pass two nos. then return addition of the nos. (2 + 3 = 5)

But suppose we pass string then it concatenate the string and return it. (“2” + “3” = 23)

Method overloading.

7. Dynamic Binding: The function call is for which function if it decides at run time then it is called dynamic binding. (Method overriding)

(Inheritance, Polymorphism)

8. Message Passing: Communication among the objects.

Conditional Statements:

-> simple if

-> if .. else

-> Nested if

-> else if ladder

-> switch

1. Simple if

Use: When we want to execute the statements only if cond. is true.

If(cond)

{

Statements;

}

Stat-x;

2. if …. Else

Use: When we want to execute different statement for condition is true and false.

If(cond)

{

True block stat;

}

else

{

False Block stat;

}

Stat-x;

3. nested if:

Use: When one cond. is depend upon another cond. (Nested if means one if stat within another if statement)

If(cond1)

{

If(cond2)

{

Stat1;

}

else

{

Stat2;

}

}

else

{

Stat3;

}

Stat-x;

Cond1 cond2 stat

T T 1

T F 2

F F 3

F T 3

4. else if ladder

Use: When we want to check multiple conditions one by one.

If(cond1)

{

Stat1;

}

else if(cond2)

{

Stat2;

}

……………

……………..

………………..

else

{

Stat-else;

}

Stat-x;

switch():

Use: When we want to check multiple conditions one by one at that time instead of else if ladder we may use switch.

Or To create menu driven programs.

Syntax:

switch(expr)

{

case val1:

stat1;

break;

case val2:

stat2;

break;

………………

……………….

………………

default:

stat-default;

break;

}

Stat-x;

Looping statements:

1. for()

Use: Entry controlled looping statement

When no. of iterations are known

Syntax:

for(init ; cond ; incr)

{

Body of the loop;

}

Stat-x;

for(i=1 ; i<=10 ; i++)

System.out.println(i);

1 2 3 4 ……10

I I I I …… 10 times

1 1 1 1 …… 10 times

for(i=1 ; i<=10 ; ++i)

System.out.println(i);

2. while loop

Use:1. Entry controlled looping statement

2. When no. of iterations are unknown but first value of loop variable is known.

Syntax:

while(cond)

{

Body of the loop;

}

Stat-x;

-> First it checks the cond.

-> If cond. is true then it executes the body of the loop.

-> Again check the cond.

-> Continue this process until cond. is false.

3. do … while loop

Use:1. Exit controlled looping statement

2. When no. of iterations and first value of loop variable both are unknown.

Syntax:

do

{

Body of the loop;

}

while(cond);

Stat-x;

-> First it executes the body of the loop whithout checking any kind of cond.

-> Then it checks the cond.

-> If cond. is true then it re-executes the body of the loop.

-> Again check the cond.

-> Continue this process until cond. is false.

1. Input an int no. WAP that check whether the no. is palindrome or not.
2. Input an int no. WAP that check whether the no. is armstrong or not.
3. Input an int no. WAP that check whether the no. is murphy or not.
4. Input nos. one by one until the no. is zero. WAP that print sum and average of all the nos.
5. Input nos. one by one until the no. is -999. WAP that count and print total positive nos., negative nos and zeros.
6. Input nos. one by one until the no. is zero. WAP that print min. and max. no.

break:

Use: Early exit from the loop.

break;

continue:

Use : To continue the loop with next value of the loop var.

continue;

int a; // Declaration

a=10; // Assignment

int a = 10; // Initialization

Object

Class

Abstraction

Inheritance

Polymorphism

Encapsulation

Message passing

Dynamic Binding

Inheritance:

To acquire the properties of one class into another is called “inheritance”.

The new class is known as sub class (Child Class) and the old one is known as super class (parent Class).

Inheritance provides reusability facility.

Types Of inheritance

-> Single Inheritance

-> Multiple Inheritances (Java Does Not Support)

-> Hierarchical Inheritance

-> Multi Level Inheritance

-> Hybrid Inheritance

-> Multi Path Inheritance

Super Sub Class

Private Protected Public

Private Not Inherited

Protected Private Protected Protected

Public Private Protected Public

Package:

Package is collection of classes, interfaces etc…

package keyword is useful to create package.

Import statement is useful to import the package it means it include all the utilities of package into our program.

Default package is java.lang

Error:

Compile Time Error :

Run Time Error

-> Logical Error (Debugging .)

-> Data Error (Wrong type)

-> Run Time Error

5/1 = 5

5/0.1 = 50

5/0.000001 = 500000

5/0 = NP = Infinite

* Askdask