

## Assignment No :- 21

(Q1) Why we can't create object of the class which contains pure virtual function in it?

→ In C++, a class that contains at least one pure virtual function is called an abstract class.

- we can't create an object of an abstract class because it has incomplete functionality that must be implemented by a derived class.

- we can't create the object because the vtable of that class is incomplete as vtable is a part of object. Our object also incomplete and object can't be incomplete.

- It is compulsory to provide the definition inside child class for the abstract class of parent class.

(Q2) What do you mean by pure virtual function?

→

A pure virtual function in C++ is a virtual function that must be overridden by derived classes. It is declared in base class using the = 0 syntax,

which means it has no implementation in the base class.

```
#include<iostream>
```

```
using namespace std;
```

```
class parent
```

```
{ public :
```

```
    int i, j, k; // concrete method
```

Void fun()

{ cout << "parent fun()" << endl; }

virtual void gun()

{ cout << "parent gun()" << endl; }

virtual int Addition (int no1, int no2) = 0; // abstract class

// pure virtual function

};

class child : public parent

{ public:

int a, b;

void gun()

{ cout << "child gun()" << endl; }

virtual void sun()

{ cout << "child sun()" ; }

int Addition (int no1, int no2)

{ return no1 + no2; }

};

int main()

{ parent pobj;

parent \*ptr = new child;

ptr -> fun();

ptr -> gun();

int ret = 0;

ret = ptr -> Addition (10, 20);

cout << "Addition is :" << ret << endl;

};

- In this example parent class is an abstract class because it contains pure virtual function in it.

In the parent class addition function is pure virtual means function without body.

- Inside derived class we provide the definition of addition function.

Q3) What will happen if the base class contains virtual function but under private access specifier?

→ If a virtual function is declared the private access specifier in base class, it is still inherited by derived classes, but it cannot be directly accessed or overridden by them.

- The virtual function remains part of the vtable mechanism.

- Derived class cannot override the private virtual function directly.

- However, the base class can provide a public or protected wrapper function to call the private virtual function.

Q4) What is mean by concrete method and abstract method?

→ Concrete and Abstract method both terms are commonly used in object-oriented programming especially in languages like C++, Java, Python.

### 1) Concrete Method

A concrete method is a method that has a complete implementation in a class.

It contains both a function declaration and a definition.

### 2) Abstract Method (Pure Virtual Function in C++)

It is a method that only has a declaration.

but no implementation in the base class.

- It is defined using = 0 in C++.

- The class containing at least one abstract method becomes an abstract class.

- Derived classes must provide an implementation for the abstract method.

Q5) Explain below syntax and draw its diagrammatic representation?

→ Class base

```
{ public:
```

```
    int i;
```

```
    float f;
```

```
    virtual void gun() = 0; } // pure virtual function
```

```
    virtual void sun() = 0;
```

```
    virtual void mun()
```

```
{ cout << "base mun"; }
```

```
}
```

class Derived : public base

```
{ public:
```

```
    int i;
```

```
    double d;
```

```
    void sun()
```

```
{ cout << "derived sun"; }
```

```
    void fun()
```

```
{ cout << "derived fun"; }
```

```
    void gun()
```

```
{ cout << "derived gun"; }
```

```
    virtual void mun()
```

```
{ cout << "derived mun"; }
```

```
}
```

```
int main()
{
```

```
    base *bp = NULL;
```

Derived obj;

```
    bp = & obj;
```

$\text{pt} \rightarrow \text{func();}$  → error (function not available in parent)

$\text{bp} \rightarrow \text{gun();}$  // call from derived

$\text{bp} \rightarrow \text{sun();}$

$\text{bp} \rightarrow \text{run();}$

$\text{bp} \rightarrow \text{mrun();}$  → error

return 0;

}

Base

UPTR	200
i	0
j	0

VTable

2000
→ Parent: run
→ derived: gun

Parent: run

derived: gun

derived: sun

bp

child	300	400
UPTR	400	
i	0	
f	0	
j	0	
d	0	

VTable

400
→ derived: gun
→ derived: sun
→ base: run
300

derived: gun

derived: sun

base: run

Base class

- contains two pure virtual functions (gun() & sun())
- contains one concrete virtual function : run()
- which has an implementation.
- This makes base an abstract class, so it cannot be instantiated.

### ② Derived class

- Inherits from base.
- Implements pure virtual function gun() & sun().
- Adds its own new method func() and mun().
- Overrides run(); is not necessary since it's already implemented in base.

### ③ Pointers (bp) and function calls

- base \*bp = & dobj; it allows polymorphism
- bp → gun() & bp → sun(); call derived's overridden function
- bp → run(); call base's implementation
- bp → fun(); bp → mun(); it give error because they do not exist in base class.

### ④ Draw the memory layout and draw the assembly instructions of function calls.

class base

{ public:

int i;

float f;

void fun() // 1000

{ cout << "base fun"; }

virtual void gun() // 2000

{ cout << "base gun"; }

}

class Derived : public base

{ public:

int i;

double d;

```
virtual void fun() // 3000
```

```
{ cout << "derived fun"; }
```

```
void gun() // 4000
```

```
{ cout << "derived gun"; }
```

```
virtual void sun() // 5000
```

```
{ cout << "derived sun"; }
```

```
};
```

```
int main()
```

```
{ base *bp = new Derived;
```

```
Derived dobj;
```

```
bp->gun();
```

```
return 0;
```

```
};
```

base <sup>100</sup> <sup>200</sup> <sup>200</sup> <sup>300</sup> <sup>400</sup> <sup>500</sup> <sup>4000</sup> <sup>1000</sup> <sup>2000</sup> <sup>3000</sup> <sup>4000</sup> <sup>5000</sup> <sup>6000</sup> <sup>7000</sup> <sup>8000</sup> <sup>9000</sup> <sup>10000</sup> <sup>11000</sup> <sup>12000</sup> <sup>13000</sup> <sup>14000</sup> <sup>15000</sup> <sup>16000</sup> <sup>17000</sup> <sup>18000</sup> <sup>19000</sup> <sup>20000</sup> <sup>21000</sup> <sup>22000</sup> <sup>23000</sup> <sup>24000</sup> <sup>25000</sup> <sup>26000</sup> <sup>27000</sup> <sup>28000</sup> <sup>29000</sup> <sup>30000</sup> <sup>31000</sup> <sup>32000</sup> <sup>33000</sup> <sup>34000</sup> <sup>35000</sup> <sup>36000</sup> <sup>37000</sup> <sup>38000</sup> <sup>39000</sup> <sup>40000</sup> <sup>41000</sup> <sup>42000</sup> <sup>43000</sup> <sup>44000</sup> <sup>45000</sup> <sup>46000</sup> <sup>47000</sup> <sup>48000</sup> <sup>49000</sup> <sup>50000</sup> <sup>51000</sup> <sup>52000</sup> <sup>53000</sup> <sup>54000</sup> <sup>55000</sup> <sup>56000</sup> <sup>57000</sup> <sup>58000</sup> <sup>59000</sup> <sup>60000</sup> <sup>61000</sup> <sup>62000</sup> <sup>63000</sup> <sup>64000</sup> <sup>65000</sup> <sup>66000</sup> <sup>67000</sup> <sup>68000</sup> <sup>69000</sup> <sup>70000</sup> <sup>71000</sup> <sup>72000</sup> <sup>73000</sup> <sup>74000</sup> <sup>75000</sup> <sup>76000</sup> <sup>77000</sup> <sup>78000</sup> <sup>79000</sup> <sup>80000</sup> <sup>81000</sup> <sup>82000</sup> <sup>83000</sup> <sup>84000</sup> <sup>85000</sup> <sup>86000</sup> <sup>87000</sup> <sup>88000</sup> <sup>89000</sup> <sup>90000</sup> <sup>91000</sup> <sup>92000</sup> <sup>93000</sup> <sup>94000</sup> <sup>95000</sup> <sup>96000</sup> <sup>97000</sup> <sup>98000</sup> <sup>99000</sup> <sup>100000</sup> <sup>101000</sup> <sup>102000</sup> <sup>103000</sup> <sup>104000</sup> <sup>105000</sup> <sup>106000</sup> <sup>107000</sup> <sup>108000</sup> <sup>109000</sup> <sup>110000</sup> <sup>111000</sup> <sup>112000</sup> <sup>113000</sup> <sup>114000</sup> <sup>115000</sup> <sup>116000</sup> <sup>117000</sup> <sup>118000</sup> <sup>119000</sup> <sup>120000</sup> <sup>121000</sup> <sup>122000</sup> <sup>123000</sup> <sup>124000</sup> <sup>125000</sup> <sup>126000</sup> <sup>127000</sup> <sup>128000</sup> <sup>129000</sup> <sup>130000</sup> <sup>131000</sup> <sup>132000</sup> <sup>133000</sup> <sup>134000</sup> <sup>135000</sup> <sup>136000</sup> <sup>137000</sup> <sup>138000</sup> <sup>139000</sup> <sup>140000</sup> <sup>141000</sup> <sup>142000</sup> <sup>143000</sup> <sup>144000</sup> <sup>145000</sup> <sup>146000</sup> <sup>147000</sup> <sup>148000</sup> <sup>149000</sup> <sup>150000</sup> <sup>151000</sup> <sup>152000</sup> <sup>153000</sup> <sup>154000</sup> <sup>155000</sup> <sup>156000</sup> <sup>157000</sup> <sup>158000</sup> <sup>159000</sup> <sup>160000</sup> <sup>161000</sup> <sup>162000</sup> <sup>163000</sup> <sup>164000</sup> <sup>165000</sup> <sup>166000</sup> <sup>167000</sup> <sup>168000</sup> <sup>169000</sup> <sup>170000</sup> <sup>171000</sup> <sup>172000</sup> <sup>173000</sup> <sup>174000</sup> <sup>175000</sup> <sup>176000</sup> <sup>177000</sup> <sup>178000</sup> <sup>179000</sup> <sup>180000</sup> <sup>181000</sup> <sup>182000</sup> <sup>183000</sup> <sup>184000</sup> <sup>185000</sup> <sup>186000</sup> <sup>187000</sup> <sup>188000</sup> <sup>189000</sup> <sup>190000</sup> <sup>191000</sup> <sup>192000</sup> <sup>193000</sup> <sup>194000</sup> <sup>195000</sup> <sup>196000</sup> <sup>197000</sup> <sup>198000</sup> <sup>199000</sup> <sup>200000</sup> <sup>201000</sup> <sup>202000</sup> <sup>203000</sup> <sup>204000</sup> <sup>205000</sup> <sup>206000</sup> <sup>207000</sup> <sup>208000</sup> <sup>209000</sup> <sup>210000</sup> <sup>211000</sup> <sup>212000</sup> <sup>213000</sup> <sup>214000</sup> <sup>215000</sup> <sup>216000</sup> <sup>217000</sup> <sup>218000</sup> <sup>219000</sup> <sup>220000</sup> <sup>221000</sup> <sup>222000</sup> <sup>223000</sup> <sup>224000</sup> <sup>225000</sup> <sup>226000</sup> <sup>227000</sup> <sup>228000</sup> <sup>229000</sup> <sup>230000</sup> <sup>231000</sup> <sup>232000</sup> <sup>233000</sup> <sup>234000</sup> <sup>235000</sup> <sup>236000</sup> <sup>237000</sup> <sup>238000</sup> <sup>239000</sup> <sup>240000</sup> <sup>241000</sup> <sup>242000</sup> <sup>243000</sup> <sup>244000</sup> <sup>245000</sup> <sup>246000</sup> <sup>247000</sup> <sup>248000</sup> <sup>249000</sup> <sup>250000</sup> <sup>251000</sup> <sup>252000</sup> <sup>253000</sup> <sup>254000</sup> <sup>255000</sup> <sup>256000</sup> <sup>257000</sup> <sup>258000</sup> <sup>259000</sup> <sup>260000</sup> <sup>261000</sup> <sup>262000</sup> <sup>263000</sup> <sup>264000</sup> <sup>265000</sup> <sup>266000</sup> <sup>267000</sup> <sup>268000</sup> <sup>269000</sup> <sup>270000</sup> <sup>271000</sup> <sup>272000</sup> <sup>273000</sup> <sup>274000</sup> <sup>275000</sup> <sup>276000</sup> <sup>277000</sup> <sup>278000</sup> <sup>279000</sup> <sup>280000</sup> <sup>281000</sup> <sup>282000</sup> <sup>283000</sup> <sup>284000</sup> <sup>285000</sup> <sup>286000</sup> <sup>287000</sup> <sup>288000</sup> <sup>289000</sup> <sup>290000</sup> <sup>291000</sup> <sup>292000</sup> <sup>293000</sup> <sup>294000</sup> <sup>295000</sup> <sup>296000</sup> <sup>297000</sup> <sup>298000</sup> <sup>299000</sup> <sup>300000</sup> <sup>301000</sup> <sup>302000</sup> <sup>303000</sup> <sup>304000</sup> <sup>305000</sup> <sup>306000</sup> <sup>307000</sup> <sup>308000</sup> <sup>309000</sup> <sup>310000</sup> <sup>311000</sup> <sup>312000</sup> <sup>313000</sup> <sup>314000</sup> <sup>315000</sup> <sup>316000</sup> <sup>317000</sup> <sup>318000</sup> <sup>319000</sup> <sup>320000</sup> <sup>321000</sup> <sup>322000</sup> <sup>323000</sup> <sup>324000</sup> <sup>325000</sup> <sup>326000</sup> <sup>327000</sup> <sup>328000</sup> <sup>329000</sup> <sup>330000</sup> <sup>331000</sup> <sup>332000</sup> <sup>333000</sup> <sup>334000</sup> <sup>335000</sup> <sup>336000</sup> <sup>337000</sup> <sup>338000</sup> <sup>339000</sup> <sup>340000</sup> <sup>341000</sup> <sup>342000</sup> <sup>343000</sup> <sup>344000</sup> <sup>345000</sup> <sup>346000</sup> <sup>347000</sup> <sup>348000</sup> <sup>349000</sup> <sup>350000</sup> <sup>351000</sup> <sup>352000</sup> <sup>353000</sup> <sup>354000</sup> <sup>355000</sup> <sup>356000</sup> <sup>357000</sup> <sup>358000</sup> <sup>359000</sup> <sup>360000</sup> <sup>361000</sup> <sup>362000</sup> <sup>363000</sup> <sup>364000</sup> <sup>365000</sup> <sup>366000</sup> <sup>367000</sup> <sup>368000</sup> <sup>369000</sup> <sup>370000</sup> <sup>371000</sup> <sup>372000</sup> <sup>373000</sup> <sup>374000</sup> <sup>375000</sup> <sup>376000</sup> <sup>377000</sup> <sup>378000</sup> <sup>379000</sup> <sup>380000</sup> <sup>381000</sup> <sup>382000</sup> <sup>383000</sup> <sup>384000</sup> <sup>385000</sup> <sup>386000</sup> <sup>387000</sup> <sup>388000</sup> <sup>389000</sup> <sup>390000</sup> <sup>391000</sup> <sup>392000</sup> <sup>393000</sup> <sup>394000</sup> <sup>395000</sup> <sup>396000</sup> <sup>397000</sup> <sup>398000</sup> <sup>399000</sup> <sup>400000</sup> <sup>401000</sup> <sup>402000</sup> <sup>403000</sup> <sup>404000</sup> <sup>405000</sup> <sup>406000</sup> <sup>407000</sup> <sup>408000</sup> <sup>409000</sup> <sup>410000</sup> <sup>411000</sup> <sup>412000</sup> <sup>413000</sup> <sup>414000</sup> <sup>415000</sup> <sup>416000</sup> <sup>417000</sup> <sup>418000</sup> <sup>419000</sup> <sup>420000</sup> <sup>421000</sup> <sup>422000</sup> <sup>423000</sup> <sup>424000</sup> <sup>425000</sup> <sup>426000</sup> <sup>427000</sup> <sup>428000</sup> <sup>429000</sup> <sup>430000</sup> <sup>431000</sup> <sup>432000</sup> <sup>433000</sup> <sup>434000</sup> <sup>435000</sup> <sup>436000</sup> <sup>437000</sup> <sup>438000</sup> <sup>439000</sup> <sup>440000</sup> <sup>441000</sup> <sup>442000</sup> <sup>443000</sup> <sup>444000</sup> <sup>445000</sup> <sup>446000</sup> <sup>447000</sup> <sup>448000</sup> <sup>449000</sup> <sup>450000</sup> <sup>451000</sup> <sup>452000</sup> <sup>453000</sup> <sup>454000</sup> <sup>455000</sup> <sup>456000</sup> <sup>457000</sup> <sup>458000</sup> <sup>459000</sup> <sup>460000</sup> <sup>461000</sup> <sup>462000</sup> <sup>463000</sup> <sup>464000</sup> <sup>465000</sup> <sup>466000</sup> <sup>467000</sup> <sup>468000</sup> <sup>469000</sup> <sup>470000</sup> <sup>471000</sup> <sup>472000</sup> <sup>473000</sup> <sup>474000</sup> <sup>475000</sup> <sup>476000</sup> <sup>477000</sup> <sup>478000</sup> <sup>479000</sup> <sup>480000</sup> <sup>481000</sup> <sup>482000</sup> <sup>483000</sup> <sup>484000</sup> <sup>485000</sup> <sup>486000</sup> <sup>487000</sup> <sup>488000</sup> <sup>489000</sup> <sup>490000</sup> <sup>491000</sup> <sup>492000</sup> <sup>493000</sup> <sup>494000</sup> <sup>495000</sup> <sup>496000</sup> <sup>497000</sup> <sup>498000</sup> <sup>499000</sup> <sup>500000</sup> <sup>501000</sup> <sup>502000</sup> <sup>503000</sup> <sup>504000</sup> <sup>505000</sup> <sup>506000</sup> <sup>507000</sup> <sup>508000</sup> <sup>509000</sup> <sup>510000</sup> <sup>511000</sup> <sup>512000</sup> <sup>513000</sup> <sup>514000</sup> <sup>515000</sup> <sup>516000</sup> <sup>517000</sup> <sup>518000</sup> <sup>519000</sup> <sup>520000</sup> <sup>521000</sup> <sup>522000</sup> <sup>523000</sup> <sup>524000</sup> <sup>525000</sup> <sup>526000</sup> <sup>527000</sup> <sup>528000</sup> <sup>529000</sup> <sup>530000</sup> <sup>531000</sup> <sup>532000</sup> <sup>533000</sup> <sup>534000</sup> <sup>535000</sup> <sup>536000</sup> <sup>537000</sup> <sup>538000</sup> <sup>539000</sup> <sup>540000</sup> <sup>541000</sup> <sup>542000</sup> <sup>543000</sup> <sup>544000</sup> <sup>545000</sup> <sup>546000</sup> <sup>547000</sup> <sup>548000</sup> <sup>549000</sup> <sup>550000</sup> <sup>551000</sup> <sup>552000</sup> <sup>553000</sup> <sup>554000</sup> <sup>555000</sup> <sup>556000</sup> <sup>557000</sup> <sup>558000</sup> <sup>559000</sup> <sup>560000</sup> <sup>561000</sup> <sup>562000</sup> <sup>563000</sup> <sup>564000</sup> <sup>565000</sup> <sup>566000</sup> <sup>567000</sup> <sup>568000</sup> <sup>569000</sup> <sup>570000</sup> <sup>571000</sup> <sup>572000</sup> <sup>573000</sup> <sup>574000</sup> <sup>575000</sup> <sup>576000</sup> <sup>577000</sup> <sup>578000</sup> <sup>579000</sup> <sup>580000</sup> <sup>581000</sup> <sup>582000</sup> <sup>583000</sup> <sup>584000</sup> <sup>585000</sup> <sup>586000</sup> <sup>587000</sup> <sup>588000</sup> <sup>589000</sup> <sup>590000</sup> <sup>591000</sup> <sup>592000</sup> <sup>593000</sup> <sup>594000</sup> <sup>595000</sup> <sup>596000</sup> <sup>597000</sup> <sup>598000</sup> <sup>599000</sup> <sup>600000</sup> <sup>601000</sup> <sup>602000</sup> <sup>603000</sup> <sup>604000</sup> <sup>605000</sup> <sup>606000</sup> <sup>607000</sup> <sup>608000</sup> <sup>609000</sup> <sup>610000</sup> <sup>611000</sup> <sup>612000</sup> <sup>613000</sup> <sup>614000</sup> <sup>615000</sup> <sup>616000</sup> <sup>617000</sup> <sup>618000</sup> <sup>619000</sup> <sup>620000</sup> <sup>621000</sup> <sup>622000</sup> <sup>623000</sup> <sup>624000</sup> <sup>625000</sup> <sup>626000</sup> <sup>627000</sup> <sup>628000</sup> <sup>629000</sup> <sup>630000</sup> <sup>631000</sup> <sup>632000</sup> <sup>633000</sup> <sup>634000</sup> <sup>635000</sup> <sup>636000</sup> <sup>637000</sup> <sup>638000</sup> <sup>639000</sup> <sup>640000</sup> <sup>641000</sup> <sup>642000</sup> <sup>643000</sup> <sup>644000</sup> <sup>645000</sup> <sup>646000</sup> <sup>647000</sup> <sup>648000</sup> <sup>649000</sup> <sup>650000</sup> <sup>651000</sup> <sup>652000</sup> <sup>653000</sup> <sup>654000</sup> <sup>655000</sup> <sup>656000</sup> <sup>657000</sup> <sup>658000</sup> <sup>659000</sup> <sup>660000</sup> <sup>661000</sup> <sup>662000</sup> <sup>663000</sup> <sup>664000</sup> <sup>665000</sup> <sup>666000</sup> <sup>667000</sup> <sup>668000</sup> <sup>669000</sup> <sup>670000</sup> <sup>671000</sup> <sup>672000</sup> <sup>673000</sup> <sup>674000</sup> <sup>675000</sup> <sup>676000</sup> <sup>677000</sup> <sup>678000</sup> <sup>679000</sup> <sup>680000</sup> <sup>681000</sup> <sup>682000</sup> <sup>683000</sup> <sup>684000</sup> <sup>685000</sup> <sup>686000</sup> <sup>687000</sup> <sup>688000</sup> <sup>689000</sup> <sup>690000</sup> <sup>691000</sup> <sup>692000</sup> <sup>693000</sup> <sup>694000</sup> <sup>695000</sup> <sup>696000</sup> <sup>697000</sup> <sup>698000</sup> <sup>699000</sup> <sup>700000</sup> <sup>701000</sup> <sup>702000</sup> <sup>703000</sup> <sup>704000</sup> <sup>705000</sup> <sup>706000</sup> <sup>707000</sup> <sup>708000</sup> <sup>709000</sup> <sup>710000</sup> <sup>711000</sup> <sup>712000</sup> <sup>713000</sup> <sup>714000</sup> <sup>715000</sup> <sup>716000</sup> <sup>717000</sup> <sup>718000</sup> <sup>719000</sup> <sup>720000</sup> <sup>721000</sup> <sup>722000</sup> <sup>723000</sup> <sup>724000</sup> <sup>725000</sup> <sup>726000</sup> <sup>727000</sup> <sup>728000</sup> <sup>729000</sup> <sup>730000</sup> <sup>731000</sup> <sup>732000</sup> <sup>733000</sup> <sup>734000</sup> <sup>735000</sup> <sup>736000</sup> <sup>737000</sup> <sup>738000</sup> <sup>739000</sup> <sup>740000</sup> <sup>741000</sup> <sup>742000</sup> <sup>743000</sup> <sup>744000</sup> <sup>745000</sup> <sup>746000</sup> <sup>747000</sup> <sup>748000</sup> <sup>749000</sup> <sup>750000</sup> <sup>751000</sup> <sup>752000</sup> <sup>753000</sup> <sup>754000</sup> <sup>755000</sup> <sup>756000</sup> <sup>757000</sup> <sup>758000</sup> <sup>759000</sup> <sup>760000</sup> <sup>761000</sup> <sup>762000</sup> <sup>763000</sup> <sup>764000</sup> <sup>765000</sup> <sup>766000</sup> <sup>767000</sup> <sup>768000</sup> <sup>769000</sup> <sup>770000</sup> <sup>771000</sup> <sup>772000</sup> <sup>773000</sup> <sup>774000</sup> <sup>775000</sup> <sup>776000</sup> <sup>777000</sup> <sup>778000</sup> <sup>779000</sup> <sup>780000</sup> <sup>781000</sup> <sup>782000</sup> <sup>783000</sup> <sup>784000</sup> <sup>785000</sup> <sup>786000</sup> <sup>787000</sup> <sup>788000</sup> <sup>789000</sup> <sup>790000</sup> <sup>791000</sup> <sup>792000</sup> <sup>793000</sup> <sup>794000</sup> <sup>795000</sup> <sup>796000</sup> <sup>797000</sup> <sup>798000</sup> <sup>799000</sup> <sup>800000</sup> <sup>801000</sup> <sup>802000</sup> <sup>803000</sup> <sup>804000</sup> <sup>805000</sup> <sup>806000</sup> <sup>807000</sup> <sup>808000</sup> <sup>809000</sup> <sup>810000</sup> <sup>811000</sup> <sup>812000</sup> <sup>813000</sup> <sup>814000</sup> <sup>815000</sup> <sup>816000</sup> <sup>817000</sup> <sup>818000</sup> <sup>819000</sup> <sup>820000</sup> <sup>821000</sup> <sup>822000</sup> <sup>823000</sup> <sup>824000</sup> <sup>825000</sup> <sup>826000</sup> <sup>827000</sup> <sup>828000</sup> <sup>829000</sup> <sup>830000</sup> <sup>831000</sup> <sup>832000</sup> <sup>833000</sup> <sup>834000</sup> <sup>835000</sup> <sup>836000</sup> <sup>837000</sup> <sup>838000</sup> <sup>839000</sup> <sup>840000</sup> <sup>841000</sup> <sup>842000</sup> <sup>843000</sup> <sup>844000</sup> <sup>845000</sup> <sup>846000</sup> <sup>847000</sup> <sup>848000</sup> <sup>849000</sup> <sup>850000</sup> <sup>851000</sup> <sup>852000</sup> <sup>853000</sup> <sup>854000</sup> <sup>855000</sup> <sup>856000</sup> <sup>857000</sup> <sup>858000</sup> <sup>859000</sup> <sup>860000</sup> <sup>861000</sup> <sup>862000</sup> <sup>863000</sup> <sup>864000</sup> <sup>865000</sup> <sup>866000</sup> <sup>867000</sup> <sup>868000</sup> <sup>869000</sup> <sup>870000</sup> <sup>871000</sup> <sup>872000</sup> <sup>873000</sup> <sup>874000</sup> <sup>875000</sup> <sup>876000</sup> <sup>877000</sup> <sup>878000</sup> <sup>879000</sup> <sup>880000</sup> <sup>881000</sup> <sup>882000</sup> <sup>883000</sup> <sup>884000</sup> <sup>885000</sup> <sup>886000</sup> <sup>887000</sup> <sup>888000</sup> <sup>889000</sup> <sup>890000</sup> <sup>89100</sup>

Q7) Draw the object layout of below Syntax

→ class base1

{ public :

    int i;

    float f;

    virtual void gun() = 0;

    virtual void sun() = 0;

    virtual void run() { } ; // 1000

}

class base2.

{ public :

    int i;

    float g;

    virtual void mun() = 0;

    virtual void fun() = 0;

    void func() { } ; // 2000

}

class Derived : public base1, base2

{ public :

    int i;

    double d;

    virtual void sun() // 3000

}

    virtual void fun() { } // 4000

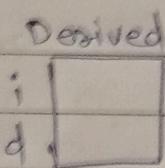
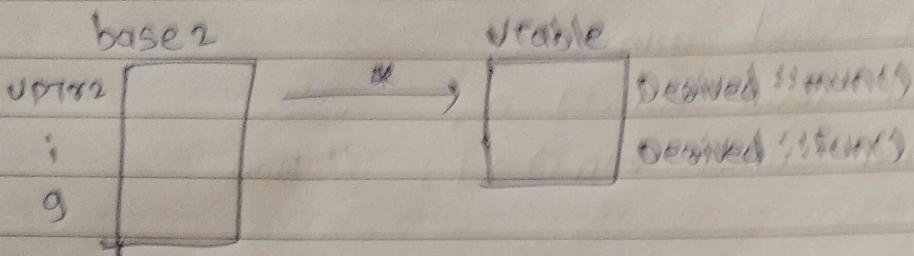
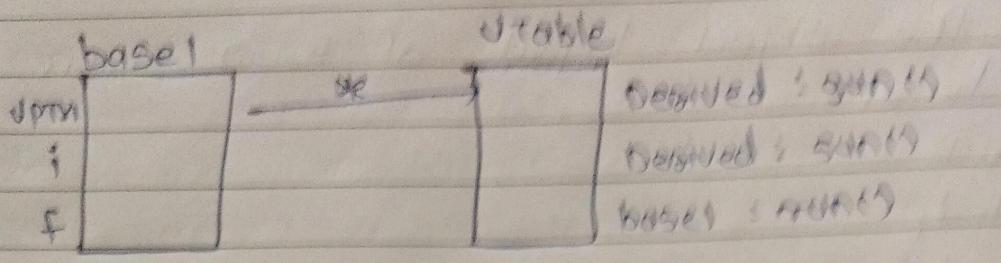
    virtual void gun() { } // 5000

    virtual void mun() { } // 6000

}

int main()

{ Derived obj;  
    return 0;



Q8) what are different ways in which we can achieve upcasting in object oriented language.

→ Upcasting is the process of converting a derived class object to a base class reference or pointer.

This allows treating the derived class object as if it were a base class object, enabling polymorphism.

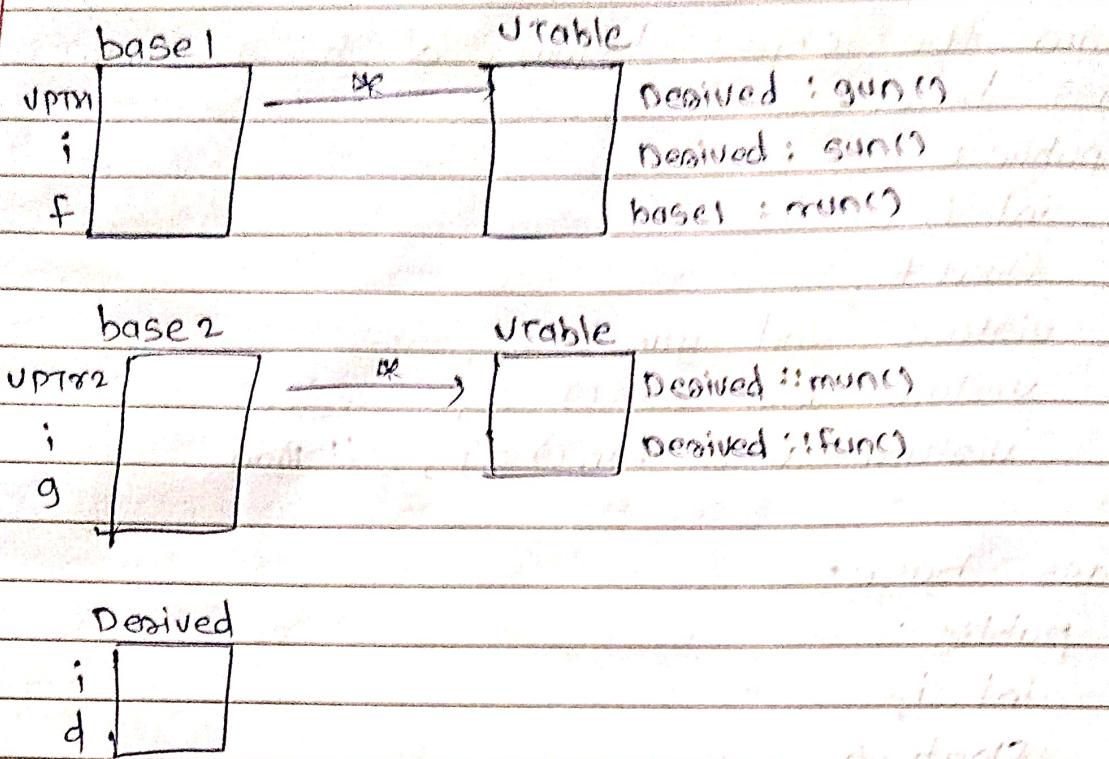
- Ways to Achieve Upcasting in C++

- ① Implicit Upcasting (Automatic)

When a derived class object is assigned to a base class reference or pointer, upcasting happens automatically.

- ② Explicit Upcasting (Using static cast)

Although upcasting is implicit, we can also do it explicitly using static cast.



Q8) What are different ways in which we can achieve upcasting in object-oriented language.

→ Upcasting is the process of converting a derived class object to a base class reference or pointer.

This allows treating the derived class object as if it were a base class object, enabling polymorphism.

- Ways to Achieve Upcasting in C++

### 1) Implicit Upcasting (Automatic)

When a derived class object is assigned to a base class reference or pointer, upcasting happens automatically.

### 2) Explicit Upcasting (using static\_cast)

Although upcasting is implicit, we can also do it explicitly using `static_cast`.

### 3) Upcasting via Function Parameters

A function that takes a base class reference or pointer can accept derived class objects.

### 4) Upcasting Using Function return type.

A function returning a base class pointer can return a derived class object.

## Q.9] What is the use of pure virtual function.

→ A pure virtual function in C++ is a virtual function that is declared in a base class but must be overridden by any derived class. It is defined using = 0 at the end of its declaration.

### • Uses of a Pure Virtual function

#### 1) Makes a class Abstract

A class containing at least one pure virtual function cannot be instantiated.

- It serves as a blueprint for derived classes.

#### 2) Supports Polymorphism

- Derived classes must provide their own implementation for the pure virtual function.

- Enables runtime polymorphism.

#### 3) Forces derived classes to implement behavior

- Ensures that all derived classes must provide their own logic.

#### 4) Provides a common interface

Allows multiple classes to follow a common interface while implementing their own logic.

Q. 10) Can we create pointers of a class which contains pure virtual function in it?

→ Yes, we can create a pointer of a class that contains a pure virtual function, but we cannot create an object of such a class.

- A class that contains at least one pure virtual function is called an abstract class.

- We cannot instantiate an abstract class directly.

- However, we can create a pointer of an abstract class and use it to point to object of derived class.