Name : Sahib Singh

Chose the correct options :

① Local variable are stored in an area called Stack

a) Heap
b) permanent storage area
c) Free Memory
[✓ d) Stack]

② Choose the corret option?

```
# include using namespace std;
  class Base {};
  class Derived : public Base;

int main()
   {
     Base *bp = new Derived;
     Derived *dp = new Base;

   }
```

a) No compiler Error
b) Compiler Error in line "Base *bp = new drived;"
[✓ c) Compiler Error in line "derived *dp = new base;"]
d) Runtime Error

3) when the inheritance is private, the private method in base class are ______ in the derived class (in C++)

a) [✓ Inaccessible]

b) Accessible

c) Protected

d) Public

4) What of the following is true?

a) The number of times destructor is called depend on number of objects created

b) destructor are called only once.

c) There can be more than one destructor In the class

[✓ d) programmer have to always call destructor at the end of the program.]

5) State True or False

Type conversion is automatic whereas type casting is explicit.

A) True

[✓ B) False]

Short answer type question ?

① 1) Explain about new and delete Keywords with code

New operator

↳ The new operator denotes a request for memory allocation on the free store. if sufficient memory is available, new operator initialize the memory and returns the address of the newly allocated and initialized memory to the printer variable.

Syntax to use new operator :

```
pointer-variable = new data-type;
```

Delete operator

↳ Since it is programmers responsibility to dellocate dynamically allocated memory, programmers are provided delete operator by C++ language

Syntax :

```
delete printer- variable;
```

③

what are constructors? why they are required? Explain different type of constructors with suitable example.

Ans : A constructor is a special type of function with no return type. Name of the constructor should be same as the name of the class.

- we define method inside the class and constructor is also defined as inside a class

A constructor is called automatically when we create an object of a class.

we required constructor

↳ * constructor does not return any value

* constructor should have a public access modifier.

Constructor Types →

1) Default constructor
2) Copy constructor
3) Static constructor
4) private constructor
5) parameterized constructor.

<u>Default constructor</u> : Default construtor is the constructor which doesn't take any argument. It has no parameter.

<u>Copy constructor</u> : A copy constructor is a member function which initializes an object using another object of the same class

<u>Paremetrized constructors</u> : it is possible to pass argument to constructor.

Typically, these arguments help initialize an object when it is created

(3) Explain the difference b/w object oriented and procedural programming language in detail.

	Procedural oriented Programming	object oriented Programming
1	Program is divided into small parts called functions	Programs is divided into small parts called objects.
2	it follows top-down approach	it follows bottom-up approach.
3	It has no ~~spe~~ access specifiers	it has access specifier like private, public and protected.
4	Adding new data and function is not easy	Adding data and function is easy
5	It is less secure because it does not have proper way for hiding data	It is more secure because it provide data hiding
6	function is more important than data	data is more important than function
7	It is based on unreal world	It is based on real world
8	C, Pascal, Basic	C++, Java, Python.

Long Answer Type question

A) Explain the type of polymorphism with code

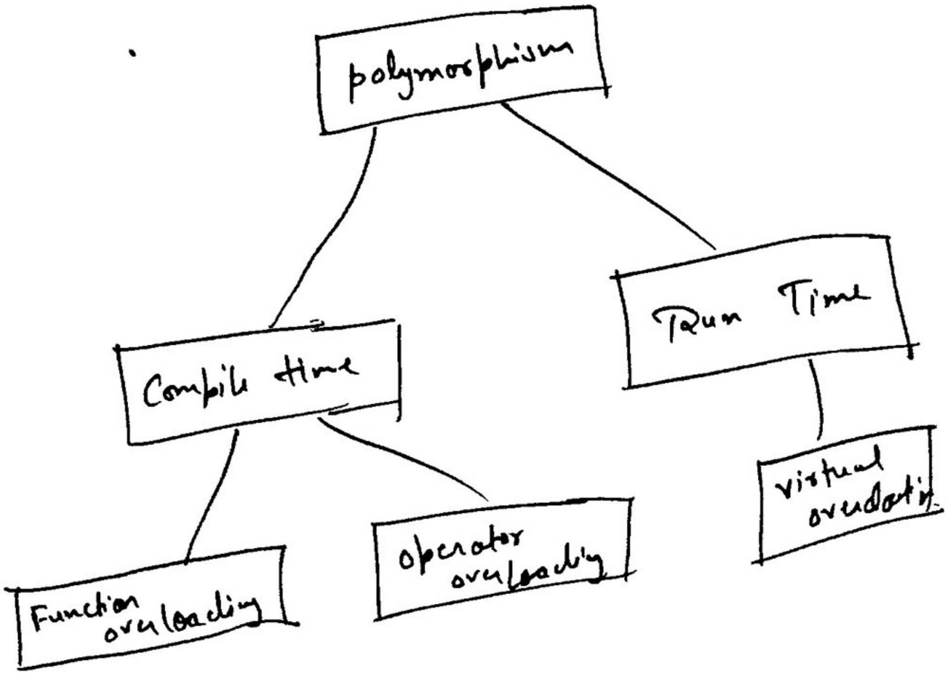
The word polymorphism means having many foorms. In simple words, we can define polymorphism as the ability of a message to be displayed in more than one form.

A real life excmple

↳ a person at the same time can have different characteristics. Like a man at the same time is a father, a husband, an employee. So the, same person posses different behaviour in different situations. This is called polymorphism.

polymorphism is divided into two types :

- Compile time polymorphism
- Run time polymorphism



(i) Compile time polymorphism :

↳ This type of polymorphism is achieved by function overloading or operator overloading

Function overloading

↳ when there are multiple functions with same name but different parameters then these functions are said to be overloaded.

Functions can be overloaded by change in number of arguments or/and change in type of arguments.

Operatr overloading :

↳ C++ also provide option to overload operators. For Example, we can make the operator ('+') for string class to concatenate two strings.

- we know that this is the additional operator whose task is to add two ~~operators~~ operands.

So, a single operator '+' when placed between integer operands, adds them and when placed between integer operands, adds them and placed between string operands, concatenates them.

② Runtime polymorphism :

↳ The runtime polymorphism is achieved when the object's method is invoked at the run time instead of compile time.