

RAILWAY PROJECT



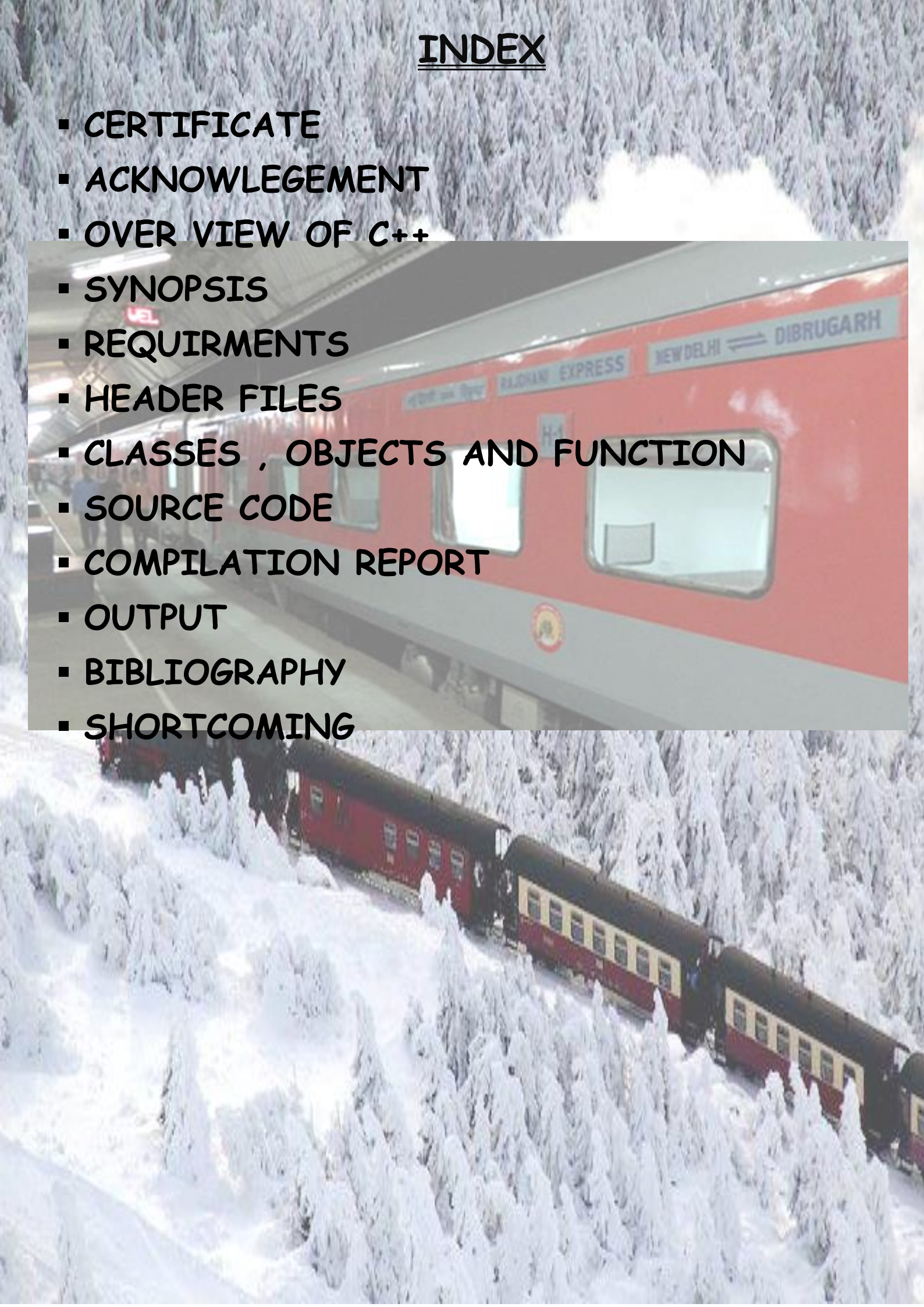
NAME: - SAMRIDDH GUPTA

ROLL NUMBER: -



INDEX

- CERTIFICATE
- ACKNOWLEDGEMENT
- OVER VIEW OF C++
- SYNOPSIS
- REQUIRMENTS
- HEADER FILES
- CLASSES , OBJECTS AND FUNCTION
- SOURCE CODE
- COMPILEATION REPORT
- OUTPUT
- BIBLIOGRAPHY
- SHORTCOMING



CERTIFICATE

This is to certify that **SAMRIDDH GUPTA** of **CLASS XII C** has done this project on "**RAILWAY RESERVATION**" under my supervision and to my complete satisfaction in the academic year of 2016-17.

MS. NITI ARORA

(PGT Computer Science)

OVER VIEW C++

C++ is a general-purpose programming language. It has imperative, object-oriented and generic programming features, while also providing facilities for low-level memory manipulation. It was designed with a bias toward system programming and embedded, resource-constrained and large systems, with performance, efficiency and flexibility of use as its design highlights. C++ has also been found useful in many other contexts, with key strengths being software infrastructure and resource-constrained applications, including desktop applications, servers (e.g. e-commerce, web search or SQL servers), and performance-critical applications (e.g. telephone switches or space probes). C++ is a compiled language, with implementations of it available on many platforms and provided by various organizations, including the Free Software Foundation (FSF's GCC), LLVM, Microsoft, Intel and IBM. C++ is standardized by the International Organization for Standardization (ISO), with the latest standard version ratified and published by ISO in December 2014 as ISO/IEC 14882:2014 (informally known as C++14). The C++ programming language was initially standardized in 1998 as ISO/IEC 14882:1998, which was then amended by the C++03, ISO/IEC 14882:2003, standard. The current C++14 standard supersedes these and C++11, with new features and an enlarged standard library. Before the initial standardization in 1998, C++ was developed by Bjarne Stroustrup at Bell Labs since 1979, as an extension of the C language as he wanted an efficient and flexible language similar to C, which also provided high-level features for program organization. Many other programming languages have been influenced by C++, including C#, D, Java, and newer versions of C (after 1998).

Objects:

C++ introduces object-oriented programming (OOP) features to C. It offers classes, which provide the four features commonly present in OOP (and some non-OOP) languages: abstraction, encapsulation, inheritance, and polymorphism. One distinguishing feature of C++ classes compared to classes in other programming languages is support for deterministic destructors, which in turn provide support for the Resource Acquisition is Initialization (RAII) concept.

Classes:

Classes are data types based on which objects are created. Objects with similar properties and methods are grouped together to form a Class. Thus a Class represents a set of individual objects. Characteristics of an object are represented in a class as Properties. The actions that can be performed by objects become functions of the class and are referred to as Methods.

Inheritance:

Inheritance is the process of forming a new class from an existing class or *base class*. The base class is also known as *parent class* or *super class*, the new class that is formed is called *derived class*. Derived class is also known as a *child class* or *sub class*. Inheritance helps in reducing the overall code size of the program, which is an important concept in object-oriented programming.

Data Abstraction:

Data Abstraction increases the power of programming language by creating user defined data types. Data Abstraction also represents the needed information in the program without presenting the details.

Data Encapsulation:

Data Encapsulation combines data and functions into a single unit called class. When using Data Encapsulation, data is not accessed directly; it is only accessible through the functions present inside the class. Data Encapsulation enables the important concept of data hiding possible.

Polymorphism:

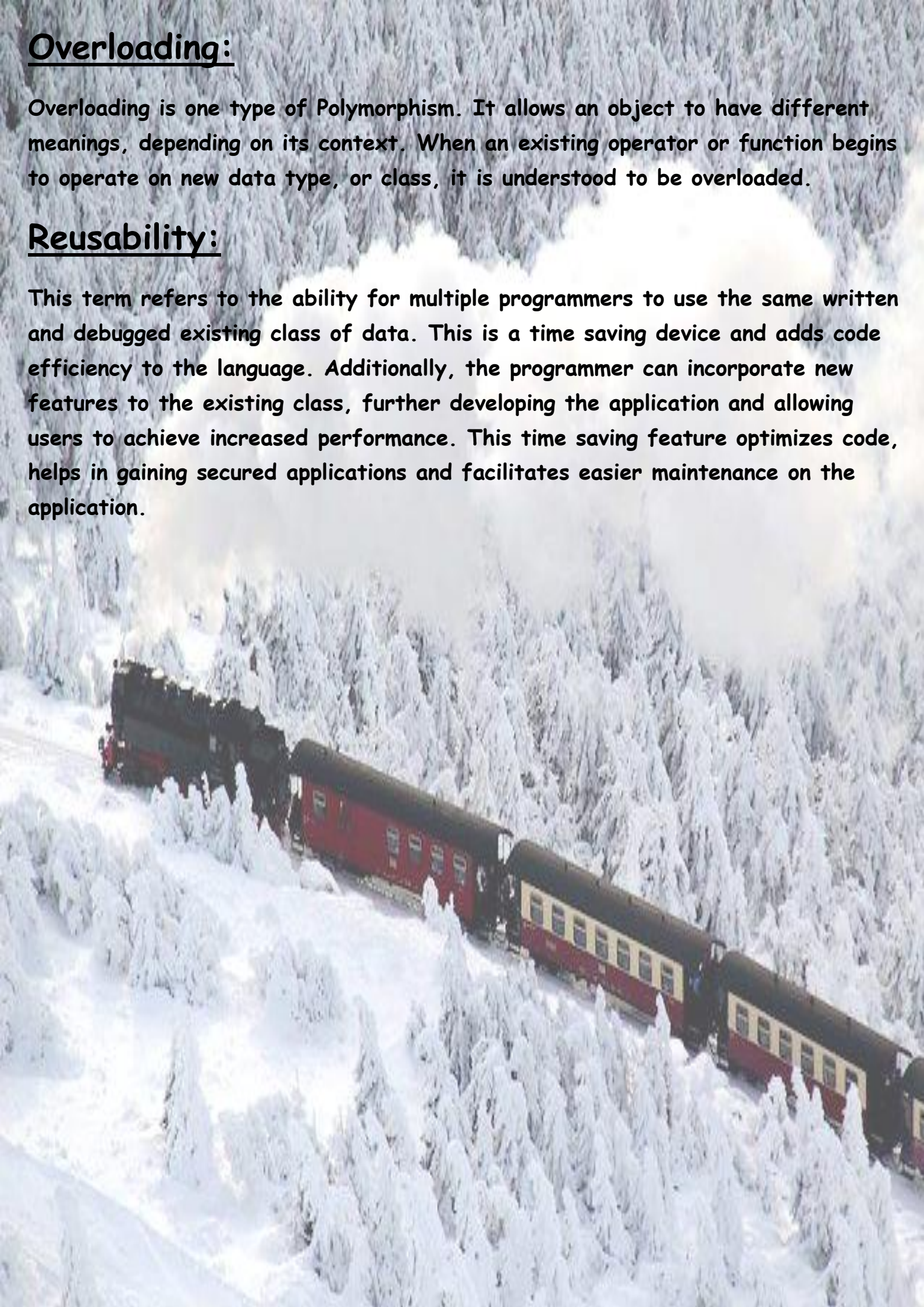
Polymorphism allows routines to use variables of different types at different times. An operator or function can be given different meanings or functions. Polymorphism refers to a single function or multi-functioning operator performing in different ways.

Overloading:

Overloading is one type of Polymorphism. It allows an object to have different meanings, depending on its context. When an existing operator or function begins to operate on new data type, or class, it is understood to be overloaded.

Reusability:

This term refers to the ability for multiple programmers to use the same written and debugged existing class of data. This is a time saving device and adds code efficiency to the language. Additionally, the programmer can incorporate new features to the existing class, further developing the application and allowing users to achieve increased performance. This time saving feature optimizes code, helps in gaining secured applications and facilitates easier maintenance on the application.



SYNOPSIS

Project Topic: - RAILWAY RESERVATION

Project Partner: -

+ SAMRIDDH GUPTA

+ UTSAV JAIN

Project Target: -

- + To book a ticket**
- + To cancel a ticket**
- + To see the waiting list**
- + To enter the information of traveler**

Description: -

1) The main menu allow you to

- Enter a train (Which is password projected)**
- To display the list of train**
- To book a train/reservation**
- Cancel a ticket**
- Show the list of ticket**
- TO show the details of traveler**

PROJECT REQUIREMENT

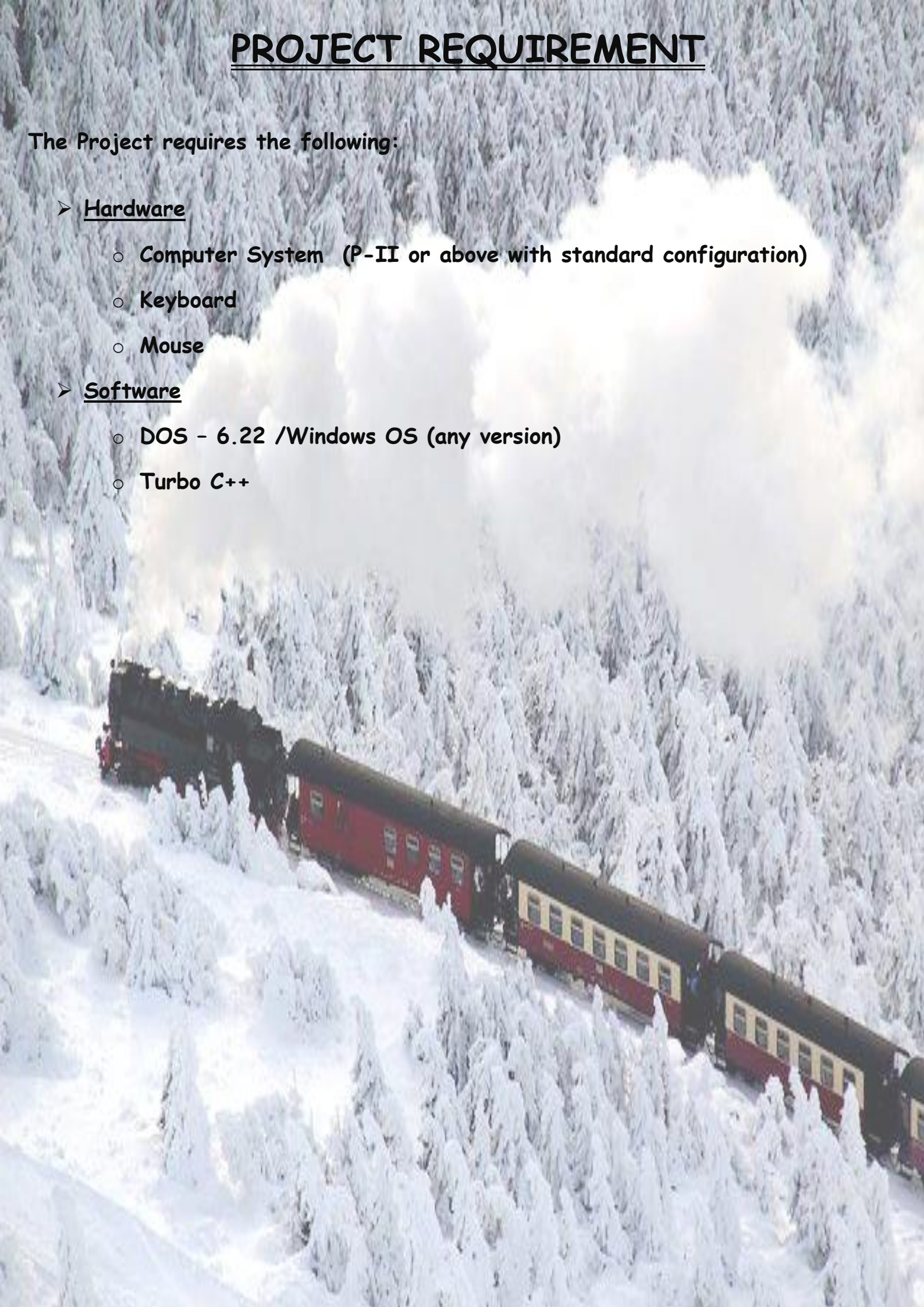
The Project requires the following:

➤ Hardware

- Computer System (P-II or above with standard configuration)
- Keyboard
- Mouse

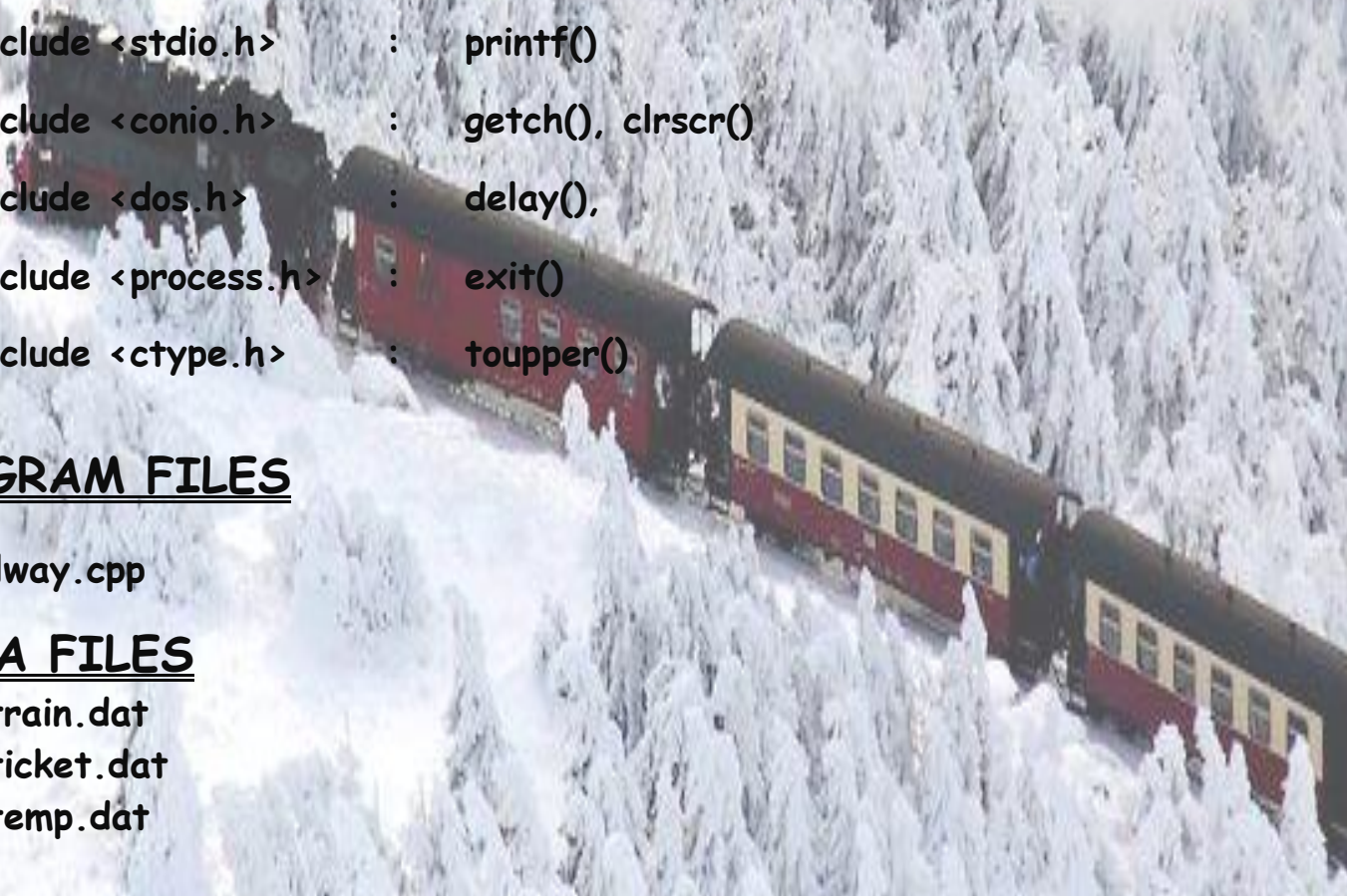
➤ Software

- DOS - 6.22 /Windows OS (any version)
- Turbo C++



Header Files Required

In computer programming, particularly in the C and C++ programming languages, a header file or include file is a file, usually in the form of source code that is automatically included in another source file by the compiler. Typically, header files are included via compiler directives at the beginning (or *head*) of the other source file. A header file commonly contains forward declarations of subroutines, variables, and other identifiers. Identifiers that need to be declared in more than one source file can be placed in one header file, which is then included whenever its contents are required. In the C and C++ programming languages, standard library functions are traditionally declared in header files

A steam locomotive pulling three passenger cars through a snowy forest.

✚ #include <fstream.h> : read(), write(), open(), close(), seekg(), seekp()
✚ #include <iostream> : cout , cin
✚ #include <string.h> : strcpy()
✚ #include <stdio.h> : printf()
✚ #include <conio.h> : getch(), clrscr()
✚ #include <dos.h> : delay(),
✚ #include <process.h> : exit()
✚ #include <ctype.h> : toupper()

PROGRAM FILES

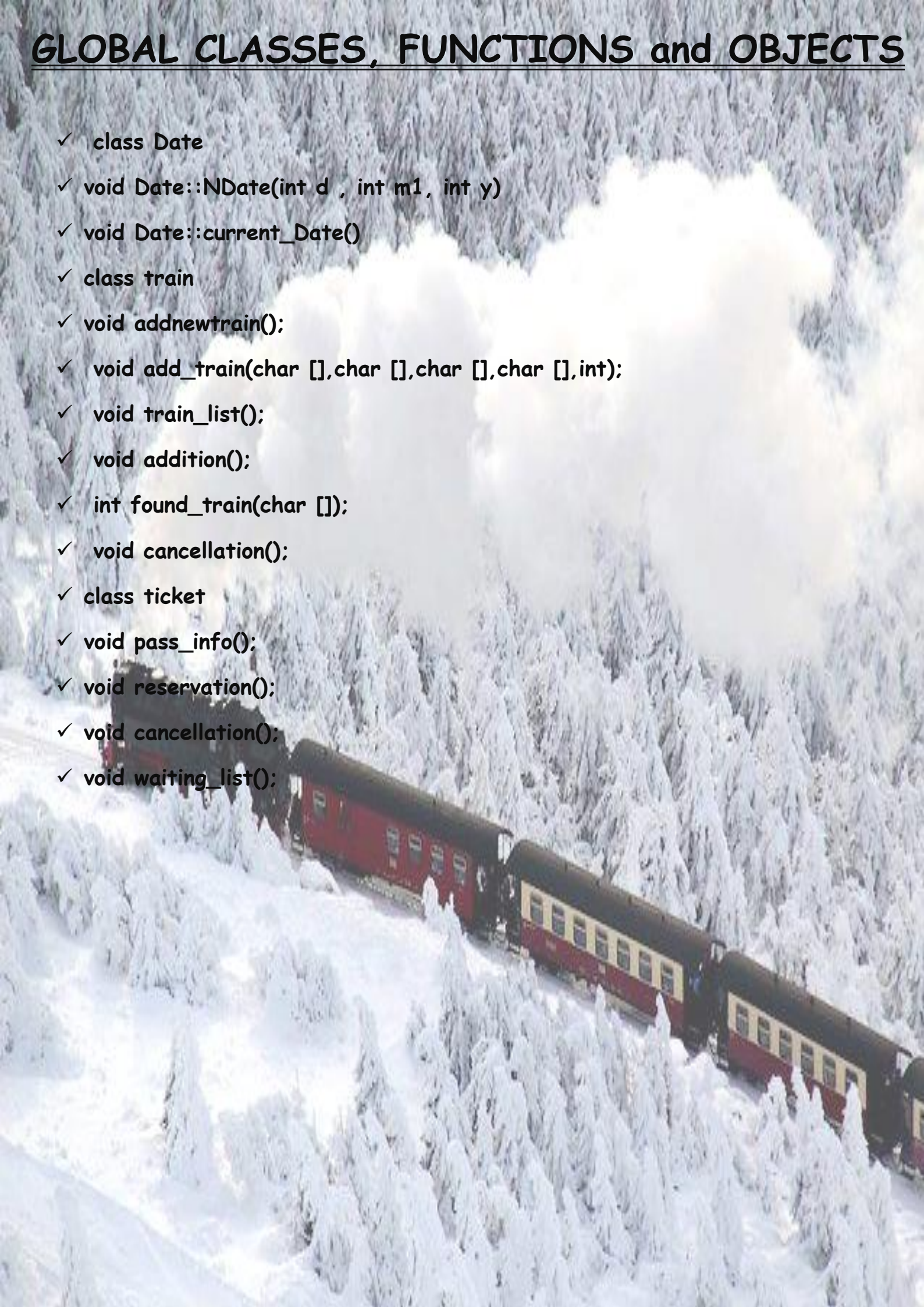
❖ Railway.cpp

DATA FILES

- ✓ train.dat
- ✓ ticket.dat
- ✓ temp.dat

GLOBAL CLASSES, FUNCTIONS and OBJECTS

- ✓ `class Date`
- ✓ `void Date::NDate(int d , int m1, int y)`
- ✓ `void Date::current_Date()`
- ✓ `class train`
- ✓ `void addnewtrain();`
- ✓ `void add_train(char [],char [],char [],char [],int);`
- ✓ `void train_list();`
- ✓ `void addition();`
- ✓ `int found_train(char []);`
- ✓ `void cancellation();`
- ✓ `class ticket`
- ✓ `void pass_info();`
- ✓ `void reservation();`
- ✓ `void cancellation();`
- ✓ `void waiting_list();`



PROJECT TITLE : "RAILWAY PROJECT"

PROGRAM NAME : RAILWAYP.CPP

MADE BY : SAMRIDDH GUPTA

CLASS : XII C

ROLL No. : 27

BOARD ROLL No. :

YEAR : 2016-17



BIBLIOGRAPHY

- 1) TURBO C++ (BY AKKI)
- 2) COMPUTER SCIENCE BOOK (BY SUMITA ARORA)
- 3) Internet



The End,

ACKNOWLEDGEMENT

It is my solemn duty to place on record my sincere thanks and deep sense of gratitude to my respected teacher, Ms.Niti Arora, for her invaluable guidance, keen interest and constant encouragement for fulfillment of this project. I also want to thank my mother who always helps me while doing the project

