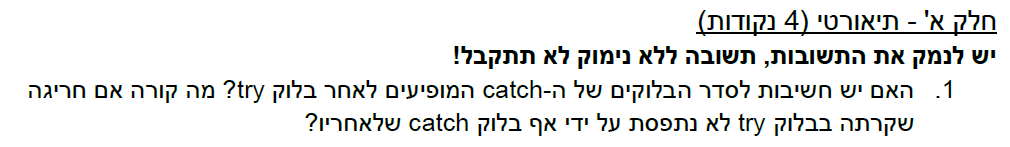
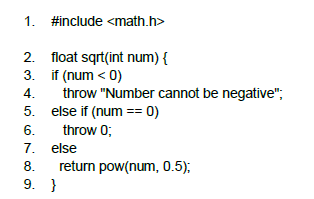
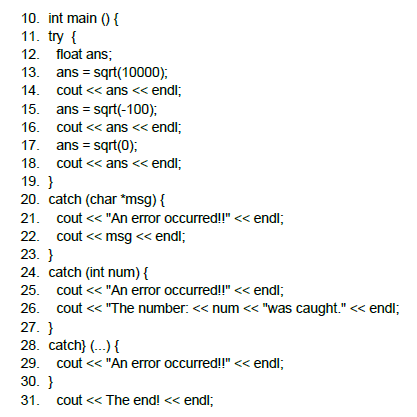
עבודת הגשה 4 – CPP

קוט אנטולי



אכן יש חשיבות לסדר של הבלוקים של ה – CATCH , כאשר יש THROW זה עובר כל CATCH ובודק אם הוא תואם ל THROW , במידה והוא לא תואם לשום CATCH הוא פשוט ימשיך רגיל , אלא אם כן יש CATCH שהוא ברירת מחדל כמו (...)





בתוכנית הנ"ל ישנתן שתי חריגות :

חריגה 1 : בשורה 15 ישנה קריאה לפונקציה עם ערך שלילי (-100) , במקרה כזה הפונקציה תזרוק מחרוזת "number cannot be negative" ולכן ה – CATCH שבשורה 20 יתפוס את ה THROW ולאחר מכן ידפיס את המחרוזת

חריגה 2 :

בשורה מספר 17 ישנה קריאה נוספת לפונקציה כשאר היא מקבלת את המספר 0 , וכתוצאה הפונקציה תחזיר ערך int שהוא 0 , ו ה catch בשורה 24 יתפוס אותו וידפיס מחרוזת מובנת עם המספר 0

מבחינתנו , רק מספר float הוא הסוג התקין שיכול לחזור מהפונקציה הזאת מכיוון ששורש לא בהכרח int

פלט תוכנית :

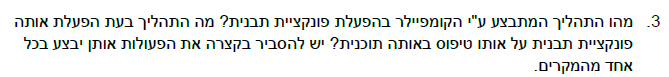
100

"An error occurred!!"

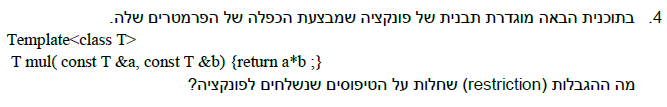
"Number cannot be negative"

"An error occurred!!"

The number 0 was caught""



בעת הקומפילציה , הקומפיילר יגש לקובץ ה – h ויגש לפונקציה של התבנית ו"תיצור פונקציה" בהתאם למה שנדרש בקוד לדוגמא , אם יתבקש סוג int או שהפרמטרים שנכנסו הו מסוג int הקומפיילר ידע לזהות ויצור את הפונקציה הזאת ורק אז תקמפל רק אותה .



ההגבלות פונקציית תבנית הנ"ל הן ש ה אובייקט T יתמוך באופרטור \*

רק אז יהיה ניתן להכפיל את ה אובייקט a מסוג T ואת האובייקט b מסוג T ולהחזיר את התשובה .