**פתרון חלק תאורטי:**

**1.** התהליך המתבצע ע"י המהדר בהפעלת פונקציית תבנית:

**א.** המהדר יוצר גרסא של פונקציה מהתבנית לפי הטיפוס הספציפי.

**ב.** המהדר מקמפל אותה.

**ג.** המהדר מפעיל אותה.

במידה ויש שימוש חוזר בפונקציה (מאותו טיפוס) הוא מבצע רק את שלב ג – מפעיל את הפונקציה.

**2.** כאשר יש יחס היררכי בין מחלקות החריגות, יש להציב טיפוס חריגה נגזר לפני הבסיס שלו, אחרת catch של הבסיס יתפוס גם את חריגת הבן (הבן הוא גם מסוג אבא).

**3.** הבעיה בקטע הקוד הוא שבהצהרת הפונקציה יש לנו throw(int,char,double) כלומר הפונקציה יכולה להיזרק עבור טיפוסים מסוג int,char,double. עבור המקרה ש test == 0 לא קיימת תפיסה (catch) ב main , כלומר אין catch עבור חריגה מסוג int. על מנת לתקן את בעיה זו נוסיף את התפיסה הבאה:

catch (int i) {

cout << "Caught integer!" << i << endl;

}

וכעת יש לנו תפיסה עבור כל סוג של חריגה שהפונקציה דורשת – int,char,double.

**4.** מכיוון שהקומפיילר יוצר מחלקה מתבנית בזמן **קומפילציה** עליו "לראות" את קוד התבנית, כולל גוף המתודות כבר בזמן קומפילציה בקובץ אחד, ולא בזמן קישור כפי שהוא מבצע לגבי פונקציות מחלקה רגילות (לא תבנית). לכן הגדרת התבנית מבוצעת כולה בקובץ ההרכזות “h”.

הקוד שמשתמש בתבנית מבצע הכללה לקובץ ע"י #include.

**5.** ההגבלות (restriction) שחלות על הטיפוסים שנשלחים לפונקציה הם:

א. אופרטור [ ]

ב. אופרטור =!

**מאור עטר - 318301231**